МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«БАШКИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**КАФЕДРА ПРОГРАММИРОВАНИЯ И ЭКОНОМИЧЕСКОЙ ИНФОРМАТИКИ**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ПО ПРОГРАММЕ БАКАЛАВРИАТА

**САГАДАТОВ РАУЛЬ РАДИКОВИЧ**

**ОПРЕДЕЛЕНИЕ ПРИЗНАКОВ ПНЕВМОНИИ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ**

|  |  |
| --- | --- |
|  | Выполнил:  Студент 4 курса очной формы обучения  Направление подготовки (специальность) 01.03.02 «Прикладная математика и информатика»  Направленность (профиль) Системное программирование и компьютерные технологии |
| Допущено к защите в ГЭК и проверено на объем заимствования:  Заведующий кафедрой  д.ф.-м.н., профессор  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Р.С. Юлмухаметов | Руководитель  к.ф.-м.н., доцент  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / К.В. Трунов |

УФА – 2021

**Содержание**

[Введение 2](#_Toc74595611)

[Постановка задачи 4](#_Toc74595612)

[1. Анализ предметной области 6](#_Toc74595613)

[1.1 Пневмония. Методы диагностики заболевания 6](#_Toc74595614)

[2. Теоретические основы нейронных сетей 11](#_Toc74595615)

[2.1 Устройство нейронных сетей 11](#_Toc74595616)

[2.3 Функции активации 15](#_Toc74595617)

[2.2 Методы обучения нейронных сетей 21](#_Toc74595618)

[2.4 Классификация моделей нейронных сетей 23](#_Toc74595619)

[2.5 Сверточные нейронные сети. 25](#_Toc74595620)

[3. Выбор инструментов разработки нейронной сети 30](#_Toc74595621)

[3.1 Язык программирования Python 30](#_Toc74595622)

[3.2 Библиотеки глубокого обучения 30](#_Toc74595623)

[4. Создание сверточной нейронной сети с использованием keras и tensorflow 33](#_Toc74595624)

[4.1 Обработка исходных данных. 33](#_Toc74595625)

[4.1.1 Аугментация данных 35](#_Toc74595626)

[4.2 Реализация модели нейронной сети. 36](#_Toc74595627)

[4.3 Обучение нейронной сети 39](#_Toc74595628)

[4.3.1 Функция потерь 39](#_Toc74595629)

[4.3.2 Оптимизатор 40](#_Toc74595630)

[4.3.3 Обучение модели нейронной сети в Keras 41](#_Toc74595631)

[5. Тестирование модели сверточной нейронной сети. 46](#_Toc74595632)

[5.1 Разработка интерфейса для нейронной сети 48](#_Toc74595633)

[Заключение 52](#_Toc74595634)

[Список литературы 53](#_Toc74595635)

Введение

На сегодняшний день машинное обучение и искусственный интеллект считаются одними из самых перспективных направлений как в ИТ-отрасли, так и во многих других. Технологии на основе нейронных сетей все активнее проникают в повседневную жизнь. Даже в, казалось бы, неочевидных вещах, таких как новостная лента в социальных сетях, новости сформированы искусственным интеллектом на основе предпочтений пользователя.

Основными задачами подобных моделей являются в первую очередь управление, классификация и прогнозирование. Главным преимуществом нейронных сетей является возможность находить и воспроизводить сложные зависимости с разным количеством переменных, а так же способность к обучению, в ходе которого происходит определение связей и зависимостей между исходными данными.

Модели машинного обучения используются в самых разных сферах деятельности человека. С помощью нейронных сетей осуществляются разные задачи такие как управление качеством изделий на производствах, эффективный поиск месторождений в нефтедобыче, обработка изображений, распознавание лиц, отпечатков, голоса, прогнозирование в экономике и многие другие. Одним из самых востребованных применений является медицина. Работа в медицинских учреждениях всегда связана со сбором, обработкой и анализом различных визуальных данных, таких как рентген, КТ и многие другие. Здесь требуются алгоритмы, способные анализировать и обрабатывать изображения. Направление искусственного интеллекта, которое занимается работой с изображениями называется Computer vision или компьютерное зрение. Данное направление считается наиболее полезным в области медицинской диагностики.

Целью данной дипломной работы является проектирование системы, на основе нейронной сети, способной распознавать признаки пневмонии на снимках рентгенограммы. Распознавание данного заболевания методами машинного обучения является актуальной задачей на данный момент. События 2020 года показали, что во многих странах диагностика данного заболевания стала проблемой, так как из-за огромного количества заболевших появилась острая нехватка специалистов.

Предметом исследования является изучение и применение актуальных библиотек для создания и обучения нейронной сети

Объектом исследования является процесс создания нейронной сети для распознавания признаков пневмонии на изображениях.

# **Постановка задачи**

Целью выпускной квалификационной работы является разработка алгоритма классификации изображений для определения пневмонии с помощью нейронных сетей.. В качестве средств разработки будут использованы такие библиотеки как Keras и Tensorflow. Для достижения цели в ходе разработки должны быть решены следующие задачи:

* обзор методов диагностики пневмонии;
* выбор метода диагностики, подходящего для классификации;
* иссдедование теоретических принципов работы нейронных сетей;
* выбор подходящего для решения поставленной задачи типа нейронных сетей;
* обзор и выбор средств для реализации модели нейронной сети;
* реализация нейронной сети, тестирование, оценка результатов;
* разработка интерфейса.

# **1. Анализ предметной области**

## **1.1 Пневмония. Методы диагностики заболевания**

Пневмония — воспаление одного или двух легких, возникшее на фоне вирусной, бактериальной или грибковой инфекции, это одно из самых опасных и тяжелых заболеваний дыхательной системы. Характеризуется выбросом жидкости в альвеолы (Альвеола — пузырьковидное образование в лёгких, оплетённое сетью капилляров. Через стенки альвеол (в лёгких человека их свыше 700 млн) происходит газообмен при дыхании), а также такими симптомами как: температура, одышка, кашель (в большинстве случаев с обильным выделением мокроты), учащение дыхания, снижение насыщения крови кислородом, что приводит к серьезным последствиям для организма. Осложнения пневмонии в форме дыхательной недостаточности, острой гипоксии грозят летальным исходом.

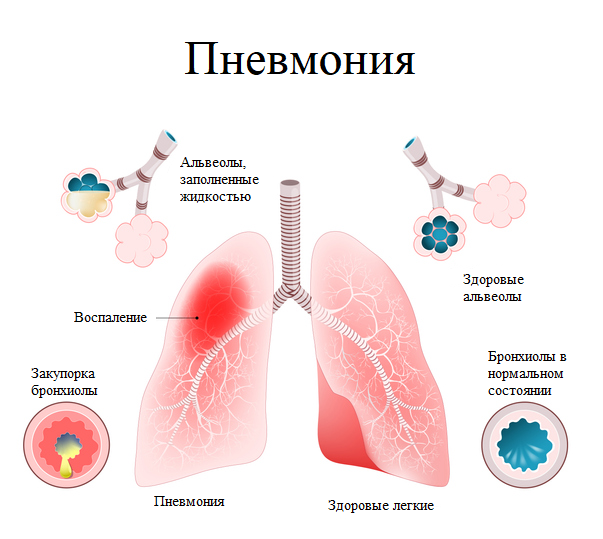


Рис.1.1 Пневмония

Пневмония может быть диагностирована разными инструментальными методами. Рассмотрим некоторые из них.

**1.Флюорография** – рентгенологическое исследование, заключающееся в фотографировании видимого изображения на флюоресцентном экране, которое образуется в результате прохождения рентгеновских лучей через тело человека и неравномерного поглощения органами и тканями организма.

С помощью флюорографии могут быть обнаружены:

* опухолевые образования и другие патологии;
* присутствие очагов воспаления;
* инородные предметы.

Преимуществами метода являются:

* минимальные затраты времени;
* минимальная стоимость исследования;
* низкое радиационное воздействие;
* широкая доступность метода и простота проведения исследований;
* не требует специальной подготовки пациента.

Недостатки метода:

* Менее точная диагностика, по сравнению с другими методами.

**2.Рентгенография грудной клетки** – это метод диагностики, позволяющий получить изображение органов грудной полости с помощью облучения рентгеновскими лучами. Различные ткани организма в зависимости от своей плотности по-разному пропускают рентгеновские лучи, а значит, по-разному отображаются на снимке (рентгенограмме).

Рентген грудной клетки используется для выявления таких патологий как:

* присутствие в легких очагов воспаления, зачастую характерных для пневмонии и туберкулеза;
* наличие опухолевых образований и отеков;
* наличие патологических скоплений газов и жидкостей;
* инородные предметы в легких, пищеводе и дыхательных путях.

Среди преимуществ данного метода можно назвать следующие факторы:

* широкая доступность метода и простота проведения исследований;
* не требует специальной подготовки пациента;
* низкая стоимость исследования.

Основные недостатки метода:

* информативность метода ниже, чем у КТ и МРТ;
* статичность изображения, невозможность проводить исследование в разных проекциях;
* большие затраты времени по сравнению с флюорографией.

**3.Компьютерная томография (КТ)** – это метод лучевой диагностики, дающий возможность послойного исследования организма, в основе которого лежит метод рентгенографии. Аппарат вращается вокруг пациента и делает серию снимков, которые затем обрабатываются и расшифровываются врачом.

С помощью КТ могут быть обнаружены:

* воспаления тканей;
* различные новообразования и отеки;
* онкологические заболевания.

Преимущества метода:

* высокая точность детализации исследуемых органов;
* результат проведения КТ-исследования – трехмерный снимок;
* полученная после КТ информация проста и понятна для восприятия.

Недостатки метода:

* относительно более высокая стоимость исследования.

**4.Магнитно-резонансная томография (МРТ)** – это диагностика внутренних органов и тканей человека с помощью ядерного магнитного резонанса. МРТ позволяет получить трехмерное изображение внутренних органов в реальном времени, что обеспечивает точную визуализацию различных структур организма.

При проведении МРТ могут быть обнаружены:

* наличие воспалительных процессов в организме;
* новообразования;
* патология структур тканей.

Преимущества МРТ:

* полученные результаты исследования характеризуются высокой точностью;
* нет ограничений по количеству МРТ процедур, в отличии от радиологических методов;
* результаты исследования в виде трехмерного снимка.

Недостатки метода:

* отсутствует возможность комплексного исследования полых органов, таких как легкие, желчный пузырь, мочевой пузырь;
* ограничения на проведение МРТ пациентам с металлическими предметами в организме и кардиостимуляторами;
* высокая стоимость проведения исследования;
* сравнительно большие временные затраты.

Хоть такие методы как КТ и МРТ достаточно точны и информативны, такие исследования все же не так доступны на данный момент. Среди рассмотренных методов диагностики оптимальными являются методы радиологических исследований, такие как рентгенография и флюорография. Эти методы широко распространены, не требуют больших затрат времени и средств. Пусть полученные с помощью них изображения сравнительно хуже, чем на МРТ и КТ, но они обладают достаточным количеством информативных признаков для классификации.

В данной главе была рассмотрена такая патология легких как пневмония, а так же методы ее диагностики.

# **2. Теоретические основы нейронных сетей**

## **2.1 Устройство нейронных сетей**

Нейронная сеть или искусственная нейронная сеть – это математическая модель, а так же ее программная или аппаратная реализация, построенная по принципу организации и функционирования биологических нейронных сетей. Нейроны являются основным типом клеток в нервной системе, которые генерируют и передают электрохимические сигналы. Они в основном общаются друг с другом с помощью нейротрансмиттеров на конкретных узлах, называемых синапсами. Большинство из них имеют две основные структуры: аксон и дендриты. Биологический нейрон состоит из ядра и отростков нервных волокон двух типов:

* дендриты – разветвленный отросток нейрона, который получает информацию;
* аксон – отросток, по которому нейрон может передавать импульс.

Аксон контактирует с дендритами других нейронов с помощью синапсов, способных влиять на силу передаваемого сигнала. Структура, состоящая из большого количества нейронов получила название биологической нейронной сети.

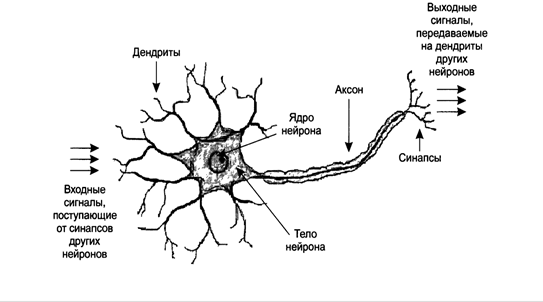


Рис. 2.1 биологический нейрон

Такой термин как «нейронная сеть» впервые появился в середине двадцатого века. Прогресс был связан с развитием нейробиологии. Первые работы, в которых был получены результаты в данном направлении, были проделаны американскими учеными У. Маккалаком и У.Питтсом. в 1943 году ими была предложена математическая модель биологического нейрона. Подобно своему биологическому прототипу, искусственный нейрон является основным элементом любой искусственной нейронной сети.

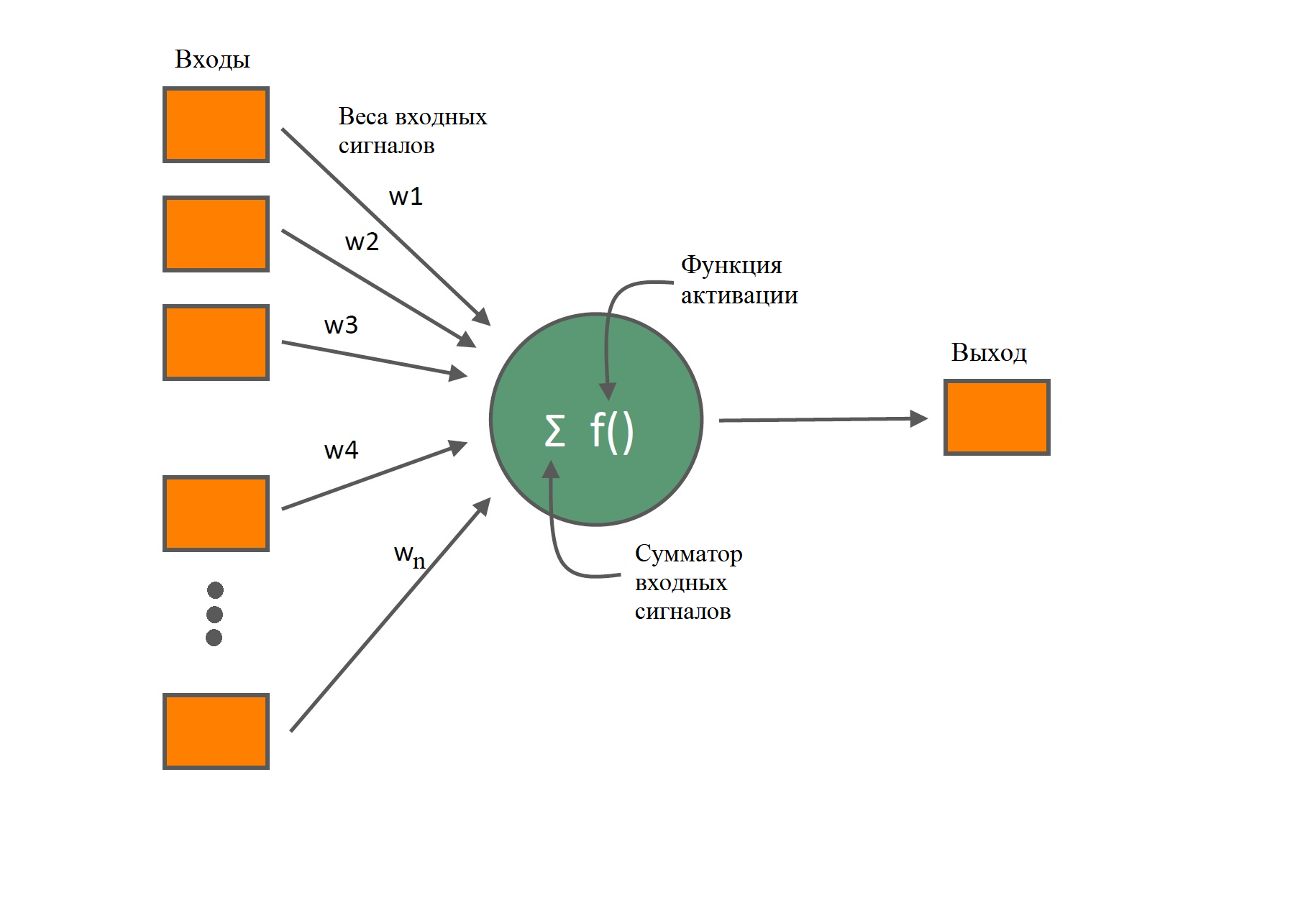


Рис. 2.2 искусственный нейрон

Математическая модель искусственного нейрона представляет собой сумматор всех входящих сигналов, применяющий к полученной сумме некоторую функцию, называемую функцией активации, затем полученный результат передается на единственный выход.

Искусственная нейронная сеть создается путем соединения между собой нейронов таким образом, что входы одних нейронов соединены с выходами других. Каждая такая связь характеризуется весом (w), являясь эквивалентом электрической проводимости в биологической нейронной сети. Таким образом, текущее состояние нейрона определяется как сумма его входов:

Где – сигналы на входах нейрона, – веса входов нейрона. - дополнительный вход и соответствующий ему вес. Это необходимо, чтобы сформировать порог чувствительности нейрона. Кроме того, иногда к выходу нейрона специально добавляют некую случайную величину, называемую сдвигом. Сдвиг можно рассматривать как сигнал на дополнительном, всегда нагруженном, синапсе.

Выход нейрона описывается функцией активации:

Данная функция определяет зависимость сигнала на выходе нейрона от текущего состояния нейрона.

Процесс обработки выходного сигнала нейрона по сути представляет собой прохождение потока данных и их преобразование. Сначала данные поступают на вход нейрона, где происходит их умножение на соответствующие весовые коэффициенты (w). Весовой коэффициент является мерой, определяющей на сколько соответствующее входное значение влияет на текущее состояние нейрона. Коэффициенты могут изменяться по мере обучения сети.

Предполагается, что нейрон мгновенно вычисляет свое выходное значение, следовательно, из таких нейронов нельзя моделировать системы с внутренним состоянием.

Недостатки модели искусственного нейрона:

* в отличии от биологических нейронов, искусственные нейроны не способны синхронно обрабатывать информацию;
* нет четких алгоритмов выбора функции активации;
* невозможно регулировать работу всей сети.

Простейшим примером нейронной сети может служить Перцептрон. Существенное влияние на развитие теории о нейронных сетях оказала работа Ф. Розенблатта «Принципы нейродинамики», изданная в 1957 году. В данной монографии он подробно описал схему перцептрона, - устройства, моделирующего процесс человеческого восприятия.

Перцептрон представляет собой передающую сеть, состоящую из генераторов сигнала трех типов: сенсорных элементов, ассоциативных элементов и реагирующих элементов.

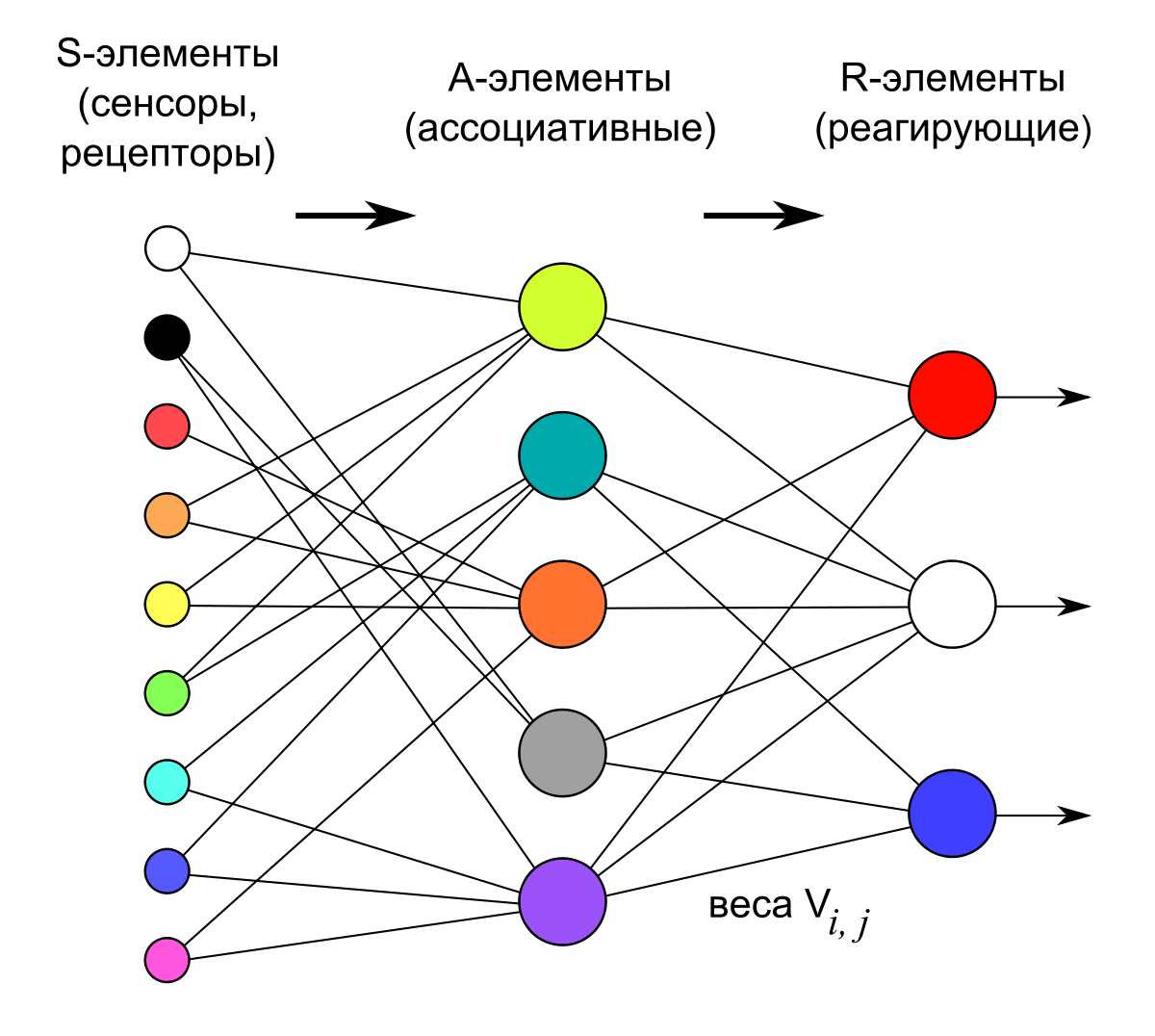


Рис. 2.3 однослойный перцептрон

Однослойный перцептрон состоит из различных элементов. Входные данные передаются с помощью сенсорных элементов ассоциативным элементам, которые выступают как связующие элементы. Ассоциативные элементы активизируются при достижении определенного числа сигналов от сенсорных элементов и передают сигнал реагирующим элементам. В зависимости от полученного сигнала результирующие элементы выдают какой-либо результат.

Теоретически число слоев данной сети может быть произвольным, однако модeль может быть ограничена ресурсами компьютера, на которых обычно реализуются мoдели нейронных сетей.

## **2.3 Функции активации**

Функция активации в искусственном нейроне определяет выходной сигнал, который определяется входным сигналом. Другими словами, с помощью нее нейрон определяет следует ли использовать сигнал или исключать его.

В искусственных нейронных сетях применяются различные функции активации. Рассмотрим наиболее популярные из них.

**1.Пороговая функция.**

Пороговая функция или функция Хевсайда – простая кусочно-линейная функция. Значение функции определяется по формуле:

Если входное значение больше порогового, то на выходе значение функции равно единице, иначе равно нулю. Она подходит для простых задач бинарной классификации, но не подходит для большего числа классов и нейронов.

График функции выглядит следующим образом:

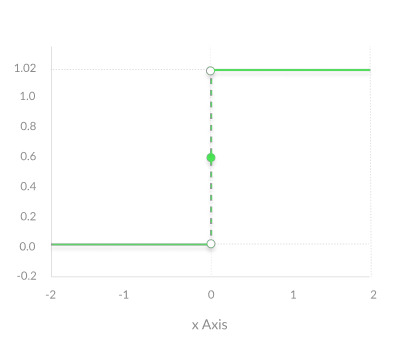


Рис. 2.4 пороговая функция активации

**2.Линейная функция активации**

Линейная функция представляет собой прямую линию, то есть результат пропорционален входному сигналу. Использование данной функции позволяет получить больше значений, чем в пороговой функции.

Значение функции определяется по формуле:

График функции:

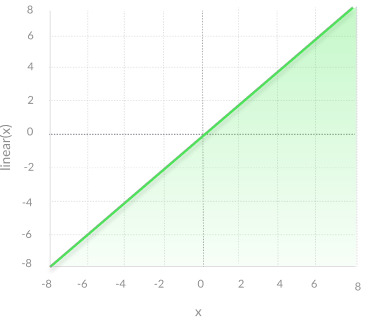


Рис. 2.5 линейная функция

Недостатки функции:

* невозможность использования метода обратного распространения ошибки, так как производная данной функции равна константе;
* имеет смысл использование в однослойных моделях сетей, так как функция линейна.

**3.Сигмоидная функция**

Сигмоидная или логистическая функция является монотонно возрастающей нелинейной функцией. Определяется по формуле:

График функции:

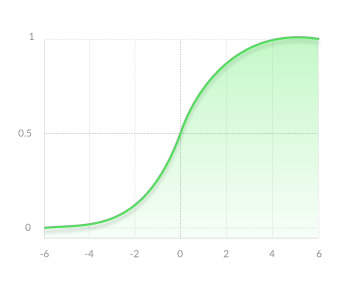


Рис. 2.6 сигмоидная функция

Очевидно, что данная функция принимает значения (0 < y < 1), что дает нормализацию выходного значения нейрона. Но по мере приближения к данным асимптотам значения у слабее реагируют на изменение аргумента. То есть производная в этих точках очень мала. Это усложняет процесс обучения с помощью градиентного спуска. Эта проблема получила название проблемы исчезающего или затухающего градиента

Недостатки функции:

* проблема затухающего градиента

**4.Гиперболический тангенс**

Гиперболический тангенс – монотонно возрастающая функция, определяется по формуле:

График функции:

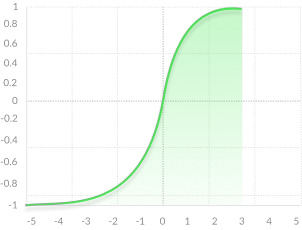


Рис. 2.7 гиперболический тангенс

Функция гиперболического тангенса является скорректированной сигмоидальной функцией, соответственно она имеет те же преимущества и недостатки, но имеет диапазон значений (-1 < y < 1). Однако производная тангенса больше, что дает большее значение градиентного спуска, следовательно и более быструю сходимость.

**5.Функция ReLU**

ReLU или Rectified Linear Unit это самая популярная функция активации в глубоком обучении. Значение ReLU равно значению аргумента, если он больше нуля, иначе равно нулю. Функция может быть описана следующей формулой:

График функции:

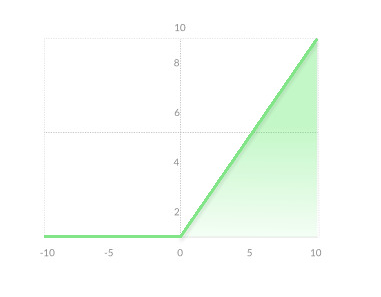


Рис. 2.8 функция ReLU

Не смотря на схожесть с линейной функцией, ReLU не является линейной, и, следовательно, ее можно применять в многослойных моделях нейронных сетей. В силу характеристик функции, использование ReLU повышает производительность обучения модели нейронной сети, так как количество включенных нейронов меньше, то есть активация разрежена. Из-за того, что значение функции для отрицательных значениях аргумента равны нулю, градиент для этих значений равен нулю, что означает игнорирование изменения в ошибке или входных данных. Такое явление называется проблемой умирающего ReLU.

Преимущества функции:

* не требователен к вычислительным ресурсам. Производная функции для отрицательных значений равна нулю, для положительных – единице;
* разреженность активации нейронов сети;

Недостатки функции:

* проблема умирающего ReLU

**6.Функция Leaky ReLU**

Одной из модификаций ReLU является функция ReLU с «утечкой» или Leaky ReLU. В отличии от обычного ReLu, график данной функции на интервале (х > 0) представляет собой не горизонтальную линию, а наклонную, с маленьким угловым коэффициентом (~0.01). Это позволяет избежать проблемы умирающего ReLU. Функция может быть описана следующей формулой:

График функции:

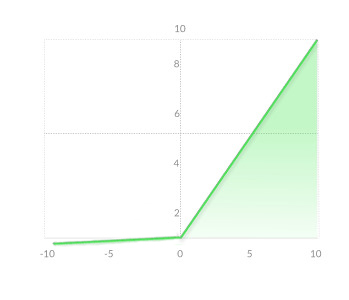


Рис. 2.9 функция Leaky ReLU

Недостатки функции:

* более требователен к вычислительным ресурсам, по сравнению со стандартной функцией, так как отрицательные значения уже не равны нулю;
* необходимо настраивать угловой коэффициент;
* на практике, результат не сильно улучшается, относительно ReLU.

Существует еще несколько модификаций ReLU функции, среди них можно отметить следующие:

* PReLU (Parametric ReLU) – угловой параметр α определяется на основе входных данных;
* RReLU (Randomized ReLU) – угловой параметр α определяется случайным образом из заданного интервалa.

## **2.2 Методы обучения нейронных сетей**

Одной из ключевых особенностей мозга как биологической нейронной сети – это способность к обучению. Постоянное обучение является естественным состоянием мозга. Что касается искусственных нейронных сетей, процесс обучения здесь может рассматриваться как настройка архитектуры сети и весов связей для наиболее эффективного решения поставленной задачи.

Впервые идею обучения нейронных сетей предложил Д. Хебб в 1949 году. Хебб предложил правила изменения веся входных сигналов нейронов в зависимости от правильности ответа. Связи нейронов, срабатывающих вместе должны усиливаться, а связи нейронов, срабатывающих отдельно от друг друга – ослабевать.

Чаще всего нейронная сеть настраивает веса по предоставленным обучающим примерам. Именно способность сети к обучению делает их более привлекательными по сравнению с другими алгоритмами.

Существующие методы обучения можно разделить на две группы: детерминированные и стохастические.

Детерминированный метод предполагает, что изменение параметров сети будет корректироваться итеративно, основываясь на ее текущих параметрах, входных значениях, фактических и желаемых выходных значений. Примером подобного метода является метод обратного распространения ошибки.

Стохастический метод обучения основывается на изменении параметров сети случайным образом. При этом сохраняются только те изменения, которые привели к улучшению результата.

В свою очередь методы обучения так же можно условно разделить на обучение с учителем и обучение без учителя.

**Обучение с учителем**

Алгоритм машинного обучения, в ходе которого система обучается с помощью примеров называется обучение с учителем. То есть модель заранее располагает требуемыми выходами сети на каждый пример в виде (Х, Y) (обучающая выборка). В процессе обучения сеть меняет свои параметры так, чтобы в итоге давать нужный ответ.

**Обучение без учителя**

Метод обучения без учителя – один из способов машинного обучения, при котором используются обычно не размеченные данные, то есть только входные данные Х, без соответствующих значений выходов Y. Иными словами сети не известна принадлежность данных к классам. Корректировка параметров сети происходит за счет поиска закономерностей, особых связях между объектами.

**Метод обратного распространения ошибки.**

Данный метод относится к методам обучения с учителем. Алгоритм предполагает два прохода по всем слоям сети: прямого и обратного. При прямом проходе подается входной вектор, в результате чего генерируется набор выходных сигналов сети, что является реакцией сети на данный образ. Во время обратного прохода все веса сети настраиваются в соответствии с правилом коррекции ошибки. Веса сети настраиваются с целью минимизации ошибки.

Математически данный метод можно описать следующим образом:

,

Где Е – функция ошибки;

– синаптический вес между нейронами i, j;

– правильные ответы сети;

– выход k-го нейрона;

– множество выходов сети.

Коррекция весов сети:

,

Где - множитель скорости движения.

Таким образом метод использует так называемый стохастический градиентный спуск, «продвигаясь» в многомерном пространстве весов в направлении антиградиента с целью достичь минимума функции ошибки. Дифференцируемость функции активации является условием использования метода. Метод является классическим и часто применяется для обучения.

## **2.4 Классификация моделей нейронных сетей**

Основной характеристикой нейронной сети является модель сети. Классифицировать нейронные сети можно по определенным параметрам таким как:

* способы обучения;
* настройка весов сети;
* тип входной информации;
* структура нейронов;
* структура модели;



Рис. 2.10 классификация нейронных сетей

Классификация нейронных сетей по организации обучения делит их на две группы:

* сети, использующие обучение с учителем;
* сети, использующие обучение без учителя.

При классификации по принципу настройки весов можно выделить:

* сети с фиксированными связями – все веса выбираются сразу;
* сети с динамическими связями – в процессе обучения происходит настройка весов.

По типу входной информации:

* аналоговая – информация подается в виде действительных чисел;
* двоичная – информация подается в виде нулей и единиц.

По типам структур нейронов:

* гомогенные – состоят из нейронов одного типа с единой функцией активации;
* Гетерогенные – состоят из нейронов с различными функциями активации.

Структура модели нейронной сети.

1. Полносвязные нейронные сети, в которых каждый нейрон связан со всеми другими нейронами сети. Входной сигнал подается всем нейронам, а выходными могут быть как все, так и некоторые выходные сигналы нейронов после нескольких тактов функционирования сети.
2. Неполносвязные нейронные сети или перцептроны могут быть описаны неполносвязным ориентированным графом. В свою очередь они подразделяются на однослойные и многослойные с прямыми, перекрестными и обратными связями.

Для задачи распознавания объектов на изображении

## **2.5 Сверточные нейронные сети.**

Сверточная нейронная сеть – это архитектура нейронных сетей, предложенная Яном Лекуном в 1988 году. Данная модель эффективно справляется с задачей распознавания образов. Сверточные нейронные сети обеспечивают устойчивость к изменением масштаба, смещениям и прочим искажениям. Структура сети – однонаправленная, т.е. без обратных связей, многослойная. Для обучения модели могут использоваться различные методы, но в основном используется метод обратного распространения ошибки. Выбор функций активации зависит от поставленной задачи. Рассмотрим подробнее устройство данной модели.

**Структура сверточной нейронной сети.**

В отличии от полносвязной сети, где каждый нейрон связан со всеми остальными и у каждой связи есть свой весовой коэффициент, сверточная нейронная сеть использует так называемую операцию свертки. Свертка предполагает собой следующий процесс: используется некоторый набор весов (ядро свертки), который перемещается по входному слою или же по самим входным данным в самом начале, формируя сигнал активации соответствующего нейрона следующего слоя. Основными видами слоев в сверточной нейронной сети являются сверточные слои, слои подвыборки (субдискретизирующие слои) и полносвязные слои.

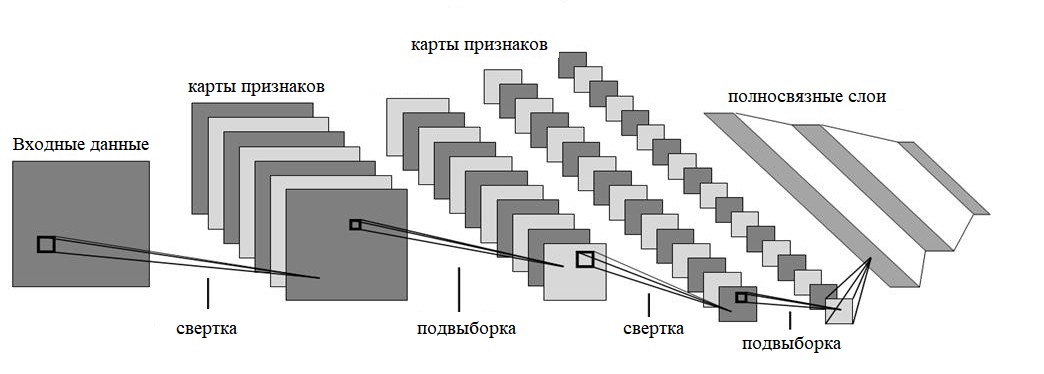


Рис. 2.11 сверточная нейронная сеть

**1.Сверточный слой**

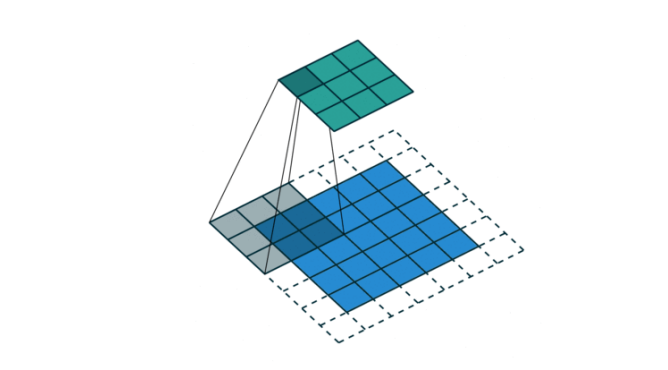
Сверточный слой представляет из себя реализацию операции свертки к входам слоя. Изображение условно разбивается на определенное количество фрагментов, пересекающихся между собой. Для поиска определенного объекта каждый фрагмент обрабатывается определенной матрицей, состоящей из весовых коэффициентов, так называемым ядром свертки. Значения интенсивности пикселя умножается на соответствующий элемент ядра свертки, затем суммируется. Скалярный результат свертки попадает на функцию активации. На выходе данного слоя получается меньшее изображение, содержащее самые выделяющиеся участки – карта признаков. Для поиска различных признаков используется несколько ядер свертки, соответственно и карт признаков на выходе будет больше. Все веса ядра свертки являются обучаемыми параметрами. 

Рис. 2.12 операция свертки

**2.Слой подвыборки**

За каждым слоем свертки следует вычислительный слой, осуществляющий локальное усреднение и подвыборку. Данный слой позволяет уменьшить размер карты признаков, сохраняя важную информацию. Преобразование происходит следующим образом: исходное изображение делится на блоки (обычно размера 2 х 2), группа пикселей в каждом блоке, проходя нелинейное преобразование, сжимаются до одного. При этом для преобразования могут использоваться разные функции, такие как максимум, среднее значение и сумма.

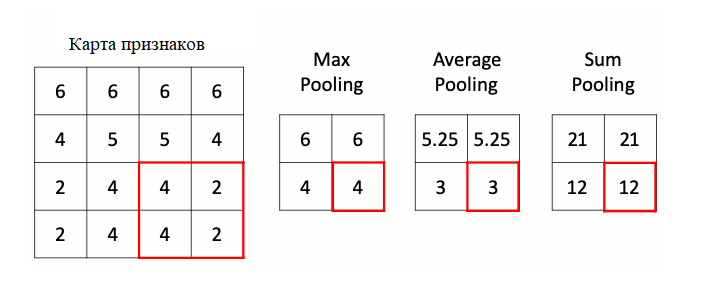


Рис. 2.13 операция подвыборки

**3.Полносвязный слой**

Данный слой представляет собой полносвязную нейронную сеть, которая, в свою очередь, может состоять из нескольких слоев. После того, как исходные данные несколько раз прошли через слои свертки и пуллинга, они перестраиваются от конкретного набора пикселей к более абстрактным картам признаков значительно меньшего размера. Как правило, перед обработкой полносвязным слоем, эти данные объединяются и обладают небольшой размерностью. Каждый нейрон в данном слое – перцептрон с нелинейной функцией активации. Результатом работы данного слоя является отнесение входных данных к какому-либо классу по найденным признакам.

Преимущества сверточной нейронной сети:

* меньшее количество настраиваемых весов, по сравнению с полносвязной сетью, так как для всего изображения используется одно ядро весов вместо использования весов для каждого пикселя. Эта особенность способствует обобщению информации, а не к запоминанию, что характерно для перцептрона;
* возможность распараллеливания вычислений;
* устойчивость к различным искажениям входных данных;
* обучение при помощи метода обратного распространения ошибки.

Из недостатков данной архитектуры можно отметить большое количество изменяемых параметров сети, таких как количество слоев, размерность ядра свертки, количество ядер. Данные настройки меняются в зависимости от задачи и располагаемой вычислительной мощности.

В данной главе были описаны основные аспекты теории нейронных сетей. Для решения задачи распознавания пневмонии на изображении, был выбран нейросетевой подход и сверточная архитектура модели.

# **3. Выбор инструментов разработки нейронной сети**

## **3.1 Язык программирования Python**

Для создания нейронных сетей могут применяться различные языки программирования. Из этого множества было решено выбрать Python.

**Python** – высокоуровневый язык программирования. Для рассматриваемой задачи, а именно – построение нейронной сети, он обладает следующими достоинствами:

* Простота, легкость в изучении. Python очень минималистичен, его синтаксис предельно прост, следовательно, код на python обладает хорошей читаемостью;
* Наличие большого количества библиотек для работы с данными, а также для создания нейронных сетей;
* Гибкость. Python является мультипарадигмальным языком программирования;
* Популярность. Язык очень востребован среди разработчиков и обладает большим сообществом.

Из недостатков языка python можно отметить относительно низкую производительность по сравнению с другими языками.

## **3.2 Библиотеки глубокого обучения**

Существует множество фреймворков и библиотек глубокого обучения. Наиболее популярные из них – это TensorFlow, Theano, caffe.

**Theano** – библиотека для python, созданная в Монреальском Университете в 2008 году. Это оптимизированная библиотека численного вычисления и работы с тензорами, предназначающаяся в качестве вычислительного «движка» для высокоуровневых интерфейсов. Работа библиотеки основана на идее вычислительных графов. Граф вычислений представляет собой конструкцию, описывающую, как будут выполняться вычисления. Theano поддерживает работу как с CPU, так и с GPU.

**Caffe** – библиотека глубокого обучения, разработанная Яньцинем Цзя в университете Беркли (BVLC). Изначально предназначалась для сверточных нейронных сетей (само название библиотеки произошло от сокращения -  «Convolution Architecture For Feature Extraction» или сверточная архитектура для извлечения признаков), caffe поддерживает множество типов машинного обучения, нацеленных в первую очередь на решении задачи распознавания образов. Одной из особенностей можно назвать возможность использования уже обученных готовых моделей нейронных сетей. Caffe поддерживает возможность использования GPU для вычислений.

**TensorFlow** – один из самых популярных фреймворков для создания и обучения нейронных сетей. Изначально TensorFlow была разработана командой Google Brain для использования в собственных проектах Google, в 2015 году система была переведена в открытый доступ. Система активно применяется для исследований в области искусственного интеллекта. Tensorflow в некотором смысле схож с Theano, работа с Tensorflow так же заключается в построении и выполнении графа вычислений. Tensorflow обладает гибкой архитектурой, которая поддерживает системы с несколькими процессорами и мобильные устройства, а так же позволяет проводить вычисления на GPU.

**Keras** – это открытая библиотека, написанная на языке python и представляющая собой удобный интерфейс для работы с такими библиотеками как Tensorflow и Theano, которые, при работе с данной библиотекой, выступают в роли вычислительного бэкенда. Keras предоставляет высокоуровневый, интуитивный набор инструментов, который упрощает проектирование нейронных сетей, вне зависимости от используемой библиотеки вычислений. Keras содержит множество реализаций применяемых блоков нейронных сетей, инструментов для работы с данными.

TensorFlow в связке с Keras – это хороший выбор для создания нейронных сетей. Tensorflow, как и Keras обладают большим сообществом и имеют подробную документацию, что делает их понятными и простыми в использовании.

Таким образом для реализации нейронной сети были выбраны следующие средства:

* язык программирования python;
* библиотеки для создания и обучения нейронных сетей Keras и tensorflow.

Помимо вышеперечисленных инструментов, в проекте так же будут использоваться вспомогательные библиотеки, такие как:

* Matplotlib, Sklearn, Seaborn для визуализации данных;
* Numpy для работы с матрицами;
* OpenCV(CV2) для считывания изображений.

# **4. Создание сверточной нейронной сети с использованием keras и tensorflow**

## **4.1 Обработка исходных данных.**

Исходные данные были взяты с сайта Kaggle. Kaggle – это публичная платформа для специалистов по машинному обучению. Проект предоставляет платформу для публикации датасетов, образования в области искусственного интелекта, а также рабочую среду разработки.

Набор данных представлен в виде изображений – рентгеновских снимков, разбитых по классам. Датасет организован тремя директориями (train, test, val) и содержит подпапки для каждой категории изображений. Всего имеется 5863 фотографии в формате JPEG.

Каталог датасета имеет следующий вид:

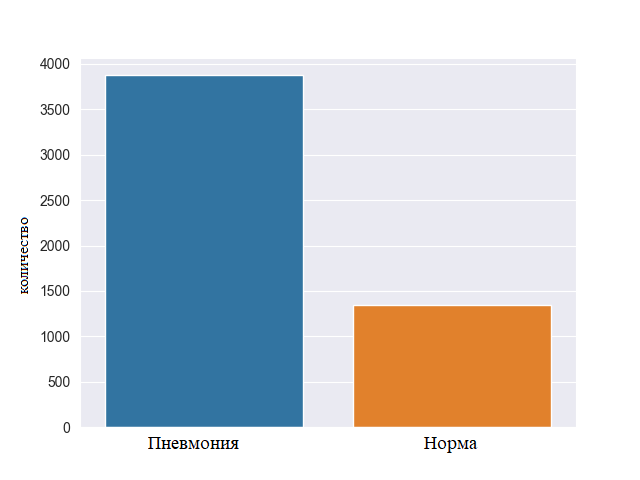
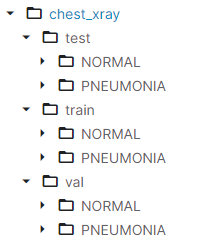


Рис. 4.1 структура данных

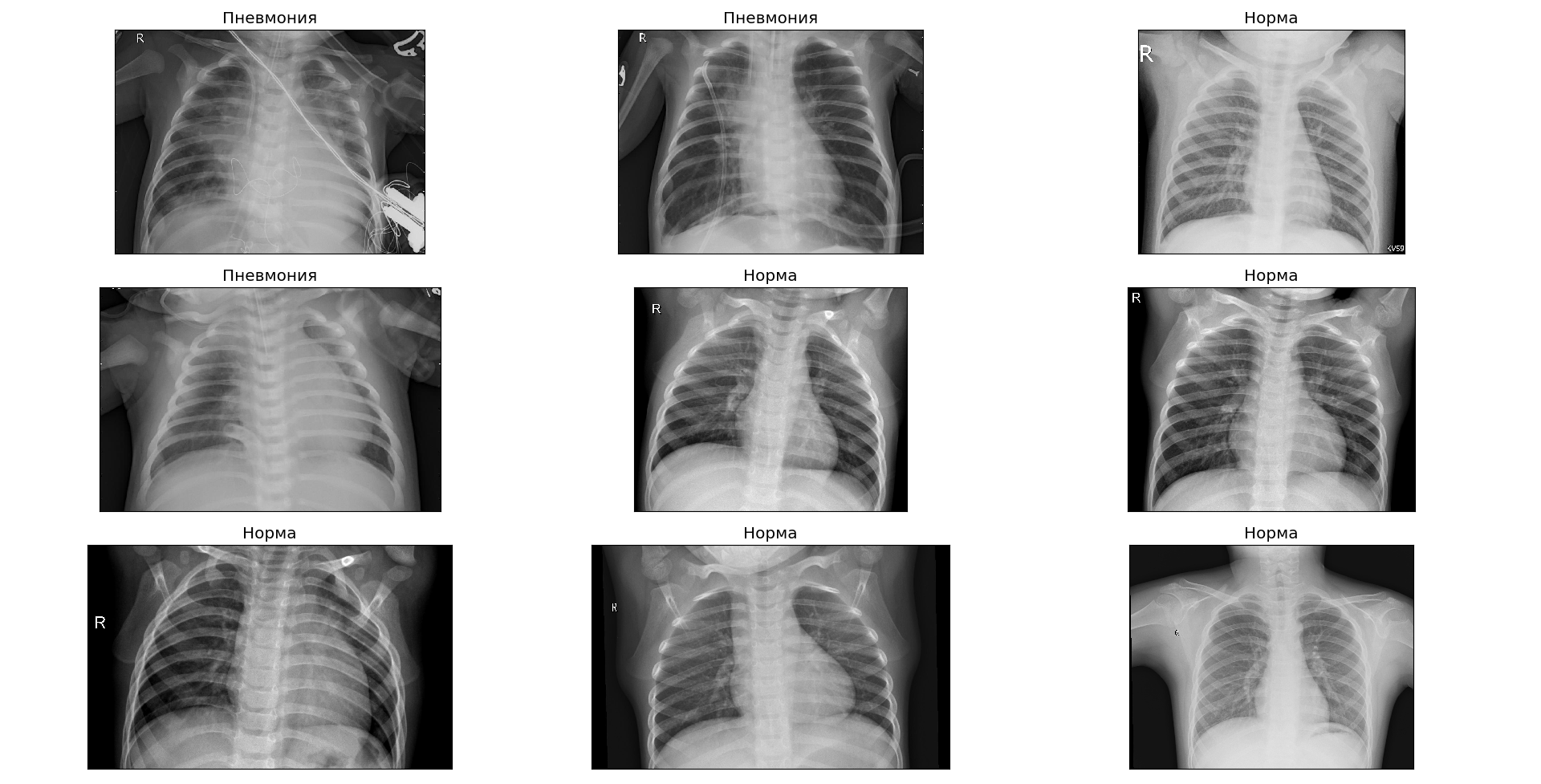


Рис. 4.2 примеры изображений

Так как исходные данные представляют из себя изображения, необходимо представить их в удобной для дальнейшей работы форме, то есть в виде массива, элементы которого соответствуют интенсивности каждого пикселя на изображении. Для этого была использована библиотека OpenCV. Метод cv2.imread() считывает изображение из файла в текущем каталоге, параметр cv2.IMREAD\_GRAYSCALE указывает, что считывание происходит в оттенках серого. Следующим шагом стоит привести полученные изображения к единому размеру с помощью cv2.resize(). Полученные массивы содержат значения от 0 до 255, что является большим диапазоном значений и для корректной работы нейронной сети требуется нормализовать данные – разделим каждый элемент на 255, получив интервал значений от 0 до 1. Далее получаем два списка, содержащих полученные изображения и соответствующие номера классов. Код данной функции представлен далее.



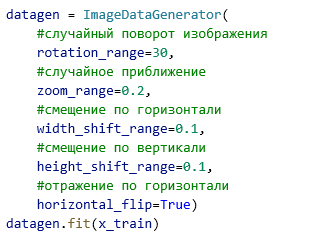
Получим необходимые данные:



### **4.1.1 Аугментация данных**

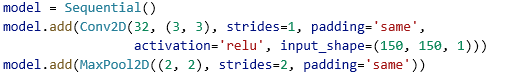
Под аугментацией данных подразумевается искусственное расширение набора данных для обучения. Помимо исходных данных применяются их модифицированные копии. Этот метод помогает справится с такой проблемой как переобучение нейронной сети. Проблема переобучения состоит в том, что обученная модель хорошо распознает данные обучающей выборки, но плохо справляется с любыми другими входными данными. Чтобы избежать этой проблемы, мы можем дополнить набор данных такими модификациями как смещение, поворот, приближение(отдаление), отражение и т.д.

Код, реализующий аугментацию:



## **4.2 Реализация модели нейронной сети.**

После того, как данные подготовлены, необходимо реализовать архитектуру сверточной нейронной сети, затем обучить ее. В Keras мы собираем слои для построения модели. Модель это обычно граф слоев. Для инициализации модели используем метод tf.keras.Sequential(). Затем добавляем слои модели с помощью model.add(), передавая тип слоя, который мы хотим добавить. В нашем случае первый слой это слой свертки Conv2D, имеющий 32 фильтра свертки 3х3, который проходит по входному изображению размером input\_shape = 150х150 пикселей с шагом strides = 1. Задаем функцию активации для данного слоя подходит функция ReLU (activation = ‘relu’). Следующим шагом будет добавление слоя подвыборки – MaxPool2D.



Слой maxpooling преобразует полученную карту признаков, уменьшая ее разрешение соответственно в два раза, то есть следующий сверточный слой получает изображения меньшего разрешения.

Мы создали первый слой сверточной нейронной сети, далее добавляем аналогичные конструкции, увеличивая количество фильтров с каждым последующим слоем.

Отличием от предыдущего слоя является дополнительный слой Dropout. Droput предназначен, чтобы избежать переобучения модели путем исключения из сети нейронов с некоторой вероятностью в процессе обучения, а затем усреднением полученных результатов.

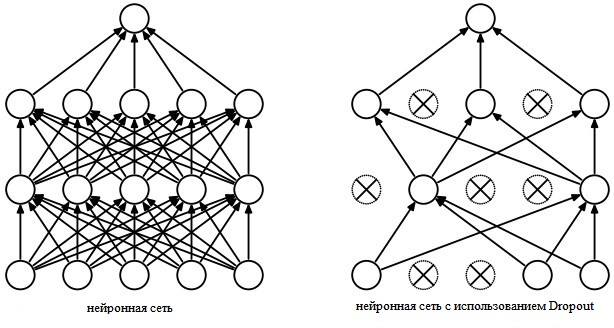
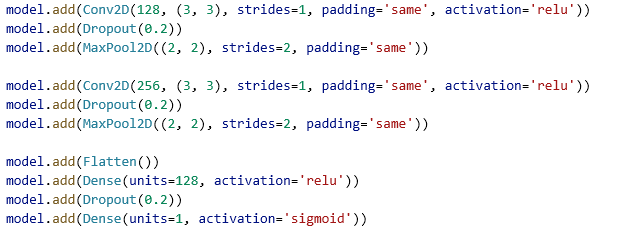


Рис 4.3 слой Dropout



Завершающим элементом является полносвязная нейронная сеть. В Keras полносвязный слой задается с помощью model.add(Dense()). Но перед тем как данные попадут на полносвязную сеть, они объединяются в единый вектор слоем model.add(Flatten()).

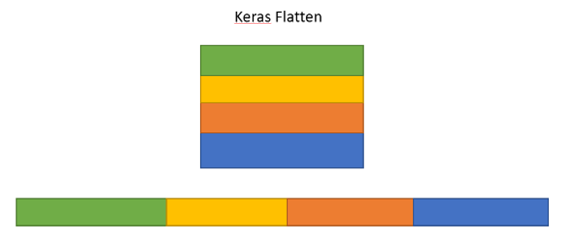


Рис 4.3 слой Flatten

Выходной слой содержит один нейрон для возможности бинарной классификации. Он использует логистическую функцию активации.

Таким образом, в качестве окончательной структуры нейронной сети была выбрана сеть с четырьмя сверточными слоями и четырьмя слоями субдескретизации, которые формируют входной вектор признаков для двух полносвязных слоев. После каждого слоя, кроме последнего использовалась функция активации ReLU, а после последнего – нелинейная сигмоидная функция. Данная структура модели была выбрана в ходе экспериментов над различными конфигурациями слоев и параметров. Описанная архитектура сверточной нейронной сети показала себя наилучшим образом на этапе тестирования.

## **4.3 Обучение нейронной сети**

Обучение нейронной сети в Keras просиходит методом обратного распространения ошибки с помощью различных модификаций градиентного спуска. Так как зачастую требуется работа с относительно большими объемами данных, которые требуют соответствующих затрат времени и вычислительных ресурсов, обычно нет возможности обрабатывать сразу все данные. Чтобы оптимизировать данный процесс, в Keras применяются такие понятия как эпохи (epochs) и партии (batches). Набор данных делится на определенное количество частей (batch) и используется для обучения. Эпоха – это итерация в процессе обучения, включающая прохождение всей обучающей выборки. Процесс обучения состоит из поиска таких весов сети, которые будут достаточно хорошими для решения конкретной задачи. Учебный процесс в данном случае является итеративным, то есть он шаг за шагом продвигается с небольшими обновлениями весов и, в свою очередь, измененяя производительность модели на каждой итерации. Итеративный процесс обучения нейронных сетей решает задачу оптимизации, которая находит параметры – веса нейронной сети, которые приводят к минимальному значению функции ошибки в наборе обучающих данных.

### **4.3.1 Функция потерь**

Функция потерь или функция ошибок, позволяет определить насколько хорошо работает нейронная сеть с набором данных. В качестве функции потерь будет использоваться бинарная перекрестная энтропия.

Бинарная перекрестная энтропия – это функция потерь, которая используется в качестве обратной связи для обучения набора весов, которую алгоритм обучения стремится свести к минимуму. Данная функция измеряет, нсколько далеко от истинного значения делается прогноз для каждого класса, а затем усредняет это значение, чтобы получить окончательное значение ошибки. Далее, с помощью оптимизаторов осуществляется поиск правильных весов, при которых функция достигает минимума.

### **4.3.2 Оптимизатор**

Оптимизаторы необходимы для повышения производительности нейронной сети. В данной модели используется оптимизатор Adam. Adam – adaptive moment estimation – алгоритм оптимизации, сочетающий в себе идею накопления движения, а так же идею малого обновления весов для признаков. Adam – особая модификация стохастического градиентного спуска. Метод вычисляет скорости обучения для каждого параметра из оценок первого и второго моментов градиентов. Данный алгоритм преобразует градиент следующим образом:

где – оценка первого момента (среднее градиентов), – оценка второго момента (средняя нецентрированная дисперсия градиентов), – градиент в момент t.

Тогда выражение для обновления параметров будет иметь следующий вид:

Где – шаг обучения, коэффициент сглаживания для предотвращения деления на 0.

Adam – один из самых популярнов алгоритмов оптимизации, по той причине, что он достигает хороших результатов относительно быстро.

### **4.3.3 Обучение модели нейронной сети в Keras**

Итак, когда данные подготовлены, а модель нейронной сети определена, можно приступать к процессу обучения модели. Для начала необходимо скомпилировать модель. Компиляция определяет функцию потерь, оптимизатор градиентного спуска и метрики.



Здесь “Adam” – оптимизатор градиентного спуска, практика показала, что этот алгоритм является наиболее эффективным для данной задачи.

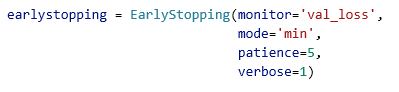
“binary\_crossentropy” – функция ошибок – бинарная перекрестная энтропия. Такая функция используется для бинарной классификации.

“metrics = [‘accuracy’]” – это метрика, по которой вычисляется точность распознавания, в данном случае это доля правильных ответов.

Для инициализации процесса обучения используется метод model.fit(), которому передаются следующие параметры:

* тренеровочные примеры и соответствующие им метки классов (x\_train, y\_train);
* количество эпох обучения;
* данные для валидации (некоторое количество тестовых примеров, для отслеживания предполагаемой точности не на тренировочной выборке);
* обратные вызовы (callbacks).

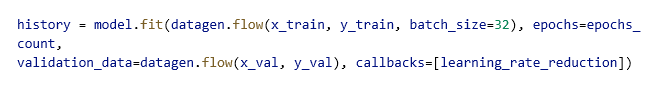
Для определения необходимого количества эпох был использован обратный вызов EarlyStopping(). При заданных параметрах, а именно, мониторинг – значение функции потерь (“monitor='val\_loss'”), ожидание – 5 (”patience=5”), обучение будет завершено, когда значение функции потерь не будет уменьшаться на протяжении пяти эпох. С помощью данной функции было выбрано необходимое количество эпох для обучения – 13.



Learning rate reduction – обратный вызов, замедляющий коэффициент скорости обучения, в данном случае (“monitor='val\_accuracy', patience=2”) при условии, что значение точности на валидационной выборке будет неизменным на протяжении двух эпох. “Factor = 0.3” определяет на какое значение будет уменьшен данный коэффициент, а “min\_lr = 0.0000001” – нижняя граница коэффициента.



Запускаем обучение модели:



Обучение будем выполнять пакетами по 32 примера (“batch\_size = 32”), количество эпох – 13. Тренировка модели выглядит следующим образом:

Рис. Обучение нейронной сети в Keras

Анализ полученных результатов показывает, что использование полученной нейронной сети в комбинации с разработанными алгоритмами предобработки исходных данных позволяет говорить об успешной работе при распознавании признаков пневмонии на снимках рентгенограммы.

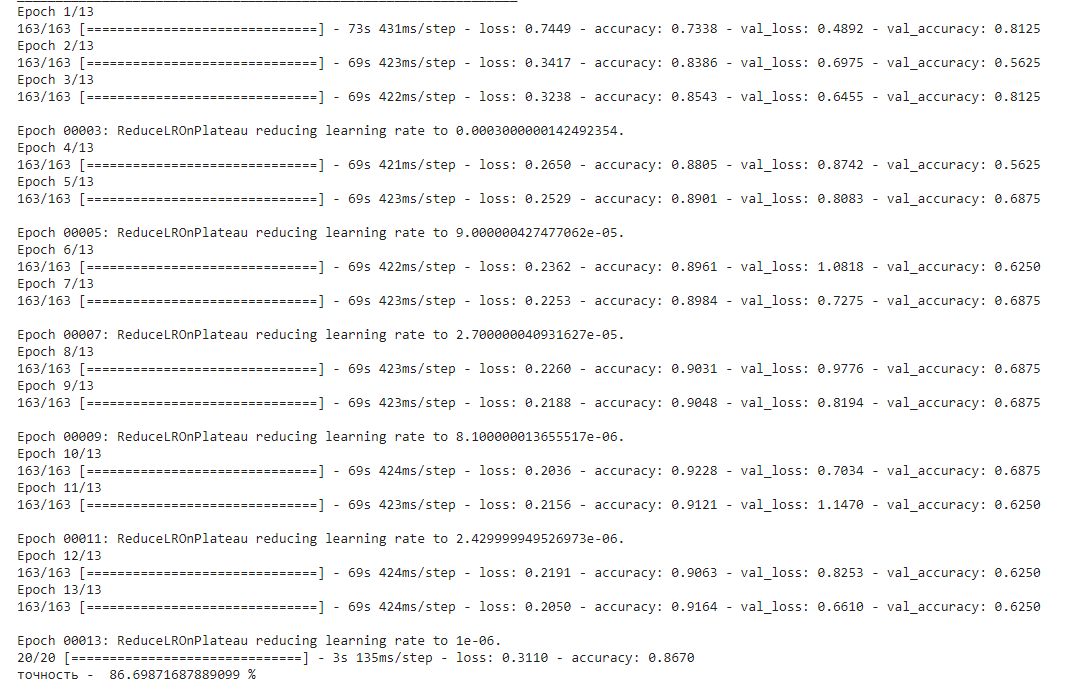
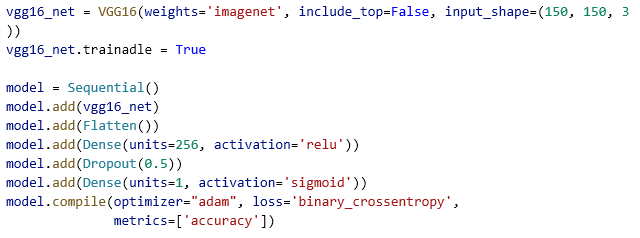
Для оценки и сравнения результатов было решено параллельно воспользоваться популярной моделью VGG-16. Архитектура данной нейронной сети представлена ниже.



Рис 4.4 Архитектура сети VGG-16

Как видно на изображении, модель состоит из 13 сверточных слоев, 5 слоев пуллинга и трех полносвязных слоев. Данная архитектура использует множество фильтров 3 х 3, в слоях подвыборки выбирается максимальное значение из квадрата 2 х 2, а все скрытые слои используют функцию активации ReLU.

VGG16 — одна из самых знаменитых моделей, отправленных на соревнование ILSVRC-2014. Она является улучшенной версией AlexNet, в которой заменены большие фильтры (размера 11 и 5 в первом и втором сверточном слое, соответственно) на несколько фильтров размера 3х3, следующих один за другим. VGG16 была разработана для классификации 1000 категорий на датасете imageNET. Для того, чтобы адаптировать данную модель для бинарной классификации, из модели была взята только сверточная часть, к которой были добавлены полносвязные слои с сигмойдной функцией активации. Стоит отметить, что VGG16 работает с цветными изображениями, то есть использует три канала, поэтому, так как тренировочный набор состоит из черно-белых изображений, было решено продублировать каналы, для того чтобы сеть могла с ними работать.



# **5. Тестирование модели сверточной нейронной сети.**

Для проведения проверки точности обученной модели в Keras существует метод “evaluate()”, на вход которому передаются элементы тестового набора данных.



Результатом работы метода являются значения точности и ошибки на тестовой выборке.



Достигнута точность распознавания в 91.2%, что является вполне хорошим результатом.

Для оценки точности модели рассмотрим возможные ошибки классификации. Прогнозы нейросети можно разделить на четыре категории:

* Истинно положительный (TP – true positive)
* Ложно положительный (FP – false positive)
* Истинно отрицательный (TN – true negative)
* Ложно отрицательный (FN – false negative)

На основе изложенных выше категорий можно составить так называемую матрицу ошибок (confusion matrix), ячейки которой представляют количество ответов, принадлежащим соответствующим категориям.

|  |  |  |  |
| --- | --- | --- | --- |
| VGG16 | Класс | Экспертная оценка | |
| Пневмония | Норма |
| Пневмония | 348(TP) | 42(FP) |
| Норма | 42(FN) | 192(TN) |

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Класс | Экспертная оценка | |
| Пневмония | Норма |
| Пневмония | 371(TP) | 19(FP) |
| Норма | 32(FN) | 202(TN) |

Рассмотрим некоторые метрики точности для оценки качества модели.

Precision (точность) – число ответов, являющихся положительными от общего количества положительно определенных ответов. Precision так же может быть описан как специфичность, то есть способность метода не давать ложноположительных ответов.

Accuracy – доля правильных ответов.

Recall (полнота) – число ответов относящихся к положительному классу. Иными словами recall демонстрирует способность модели обнаруживать данный класс. Recall так же понимают как чувствительность.

F-мера – метрика, объединяющая recall и precision. По сути это гармоническое среднее precision и recall

Проведем оценку моделей с помощью вышеописанных метрик:

Результаты сети VGG-16:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Класс | Precision | Recall | F-мера | Accuracy |
| Пневмония | 0,89 | 0,89 | 0,89 | 0,87 |
| Норма | 0,82 | 0,82 | 0,82 |

Результаты сети, рассматриваемой в данной работе:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Класс | Precision | Recall | F-мера | Accuracy |
| Пневмония | 0,92 | 0,94 | 0,93 | 0,91 |
| Норма | 0,89 | 0,87 | 0,88 |

Как видно из результатов исследования, созданная в рамках данной работы модель справляется с распознаванием пневмонии не хуже, чем сеть VGG-16. Кроме того, получившаяся модель отличается тем, что имеет меньше обучаемых параметров (1,245,313 против 16,812,353 в VGG-16), что дает ощутимую разницу в скорости обучения.

## **5.1 Разработка интерфейса для нейронной сети**

Для демонстрации работы нейронной сети на практике, требуется создать графический интерфейс, обладающий следующими функциями:

* Загрузка и отображение исследуемого изображения;
* отображение распознанного класса;
* отображение верного ответа, в случае, если он известен;
* возможность выбора верного ответа.

Для реализации программы была использована библиотека PyQT5. PyQt — набор расширений графического фреймворка Qt (c++) для языка программирования Python. Для PyQT5 так же существует приложение QT Designer, представляющая собой некий «конструктор» для создания будущего графического приложения. Это делает PyQt очень полезным инструментом для быстрого создания графического интерфейса.

Для начала нужно загрузить и скомпилировать предварительно обученную модель. Для этого воспользуемся встроенными в Keras функциями model.load(“model\_name”) и model.compile(optimizer, loss, metrics).



Класс, реализующий графический интерфейс выглядит следующим образом:



Здесь описаны все элементы интерфейса, их расположение и другие параметры.

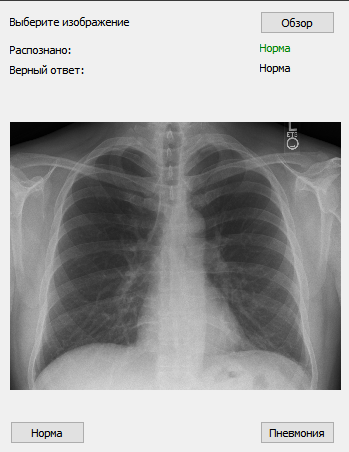
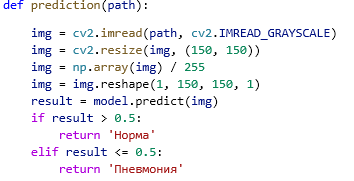


Рис. 5.1 результат работы программы

При нажатии кнопки «обзор» пользователь может выбрать изображение в открывшимся окне проводника. Затем изображение передается в функцию, преобразующую его. После преобразований полученные данные отправляются на вход модели.



Функция, возвращающая результаты работы нейронной сети:



Пользователь может внести свою оценку для текущего изображения, нажатием соответствующей кнопки («норма», «пневмония»), после чего данное изображение копируется в директорию тренировочной выборки, чтобы в последующим улучшить результаты нейронной сети.

# **Заключение**

# **Список литературы**