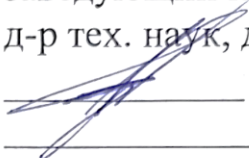


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики
Кафедра анализа данных и искусственного интеллекта

Допустить к защите
заведующий кафедрой
д-р тех. наук, доцент
 А.В. Коваленко
_____ 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ НАЛИЧИЯ
РАЗЛИЧНЫХ КАРДИОЛОГИЧЕСКИХ ПАТОЛОГИЙ ПО
КАРДИОГРАММЕ ПАЦИЕНТА

Работу выполнил(а)  Т.П. Слепцова

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность (профиль) Математическое и информационное обеспечение экономической деятельности

Научный руководитель
канд. пед. наук, доцент  В. А. Акиньшина

Нормоконтролер
канд. физ.-мат. наук, доцент  Г.В. Калайдина

Краснодар
2023

РЕФЕРАТ

Выпускная квалификационная работа 49 с., 4 ч., 29 рис., 4 табл., 9 источников.

ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ, НЕЙРОН, ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ, СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ, КОМПЬЮТЕРНОЕ ЗРЕНИЕ, КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ, КОРРЕЛЯЦИОННЫЙ АНАЛИЗ, ПРИЛОЖЕНИЕ

Объектом исследования являются искусственные нейронные сети, предназначенные для классификации изображений с помощью нейронных сетей.

Целью работы является знакомство с основными понятиями нейронных сетей, возможностями их классификации и обучения, применение свёрточных нейронных сетей, с наложением фильтров и сетки на изображение, сбор и анализ данных, построение сетки, написание приложения для прогнозирования кардиологических заболеваний.

В результате выполнения данной работы был произведен сбор данных, рассмотрены теоретические основы анализа данных, искусственных нейронных сетей, методы их обучения, а также, проанализированы перспективы применения нейронных сетей в медицине, построены математические и информационные модели для прогнозирования различных кардиологических патологий, посредством программной реализации, для одной из задач медицинской диагностики, написано приложение на языке программирования Python для предсказания наличия/отсутствия заболевания сердца, описано руководство пользователя.

СОДЕРЖАНИЕ

Введение	4
1 От биологии к информатике	5
1.1 Нейробиология и связь с искусственными нейронными сетями	5
1.2 Классификация и строение нейронных сетей	6
1.3 Принцип работы искусственных нейросетей	8
2 Машинное обучение и работа с данными	12
2.1 Задачи машинного обучения	12
2.2 Работа с данными и классификация изображений	14
2.3 Компьютерное зрение. Обработка изображений	16
2.4 Свёрточная нейронная сеть	18
2.5 Преимущества и недостатки искусственных нейронных сетей	20
3 Анализ предметной области и метода решения задачи	21
3.1 Постановка задачи	21
3.2 Построение математической модели	21
3.3 Обработка и классификация изображений с помощью Python	30
4 Решение поставленной задачи	34
4.1 Назначение системы и ее пользователи	34
4.2 Описание построения нейронной сети	35
4.3 Создание приложения	38
4.4 Руководство пользователя	40
Заключение	48
Список использованных источников	49

ВВЕДЕНИЕ

Время. В нашем быстрорастущем и развивающемся мире время играет большое значение для каждого человека. Осознание ценности времени и его быстротечности наталкивает нас на выполнение действий и получение результатов как можно скорее. Но что же может помочь нам сохранить время? Технологии. Поэтому в современном мире стали актуальны задачи автоматизации процессов деятельности человека, с помощью нейронных сетей, так называемого искусственного интеллекта.

Нейронные сети представляют собой математическую модель, в основе которой те же принципы, что и в биологических нейронных сетях.

Областей, в которых применяются искусственные нейронные сети, большое множество: от распознавания образов до прогнозирования серьезного заболевания, на основе большого количества накопленных данных.

Именно в медицине применение нейронных сетей необходимо для получения быстрого и точного результата, для повышения эффективности диагностики заболевания и прогнозирования его дальнейшего развития, а вследствие этого выбора правильной траектории лечения пациента. Из множества факторов информации о состоянии больного на момент поступления, результатов лабораторных анализов, наличия либо отсутствия хронических заболеваний, проведенных инструментальных исследований необходимо выделить именно те, которые могут повлиять на усугубление состояния больного и исключить их. Для решения этой задачи можно использовать статистический анализ и методы машинного обучения.

1 От биологии к информатике

1.1 Нейробиология и связь с искусственными нейронными сетями

Нейрон – специфическая структура, являющаяся клеткой мозга, которая получает данные, обрабатывает их и делает выводы. Этимологически нетрудно понять, что множество нейронов, связанных между собой в нервной системе, образуют нейронную сеть.

Искусственная нейронная сеть – это математическая модель, построенная по принципу организации и функционирования биологических нейронных сетей. Она представляет собой систему соединённых и взаимодействующих между собой искусственных нейронов.

Как связана нейробиология с искусственными нейронными сетями?

Нейробиология – это наука, изучающая строение, физиологию и основные функции мозга, основными объектами изучения которой являются нейроны.

Рассмотрим строение нейрона головного мозга на рисунке 1.

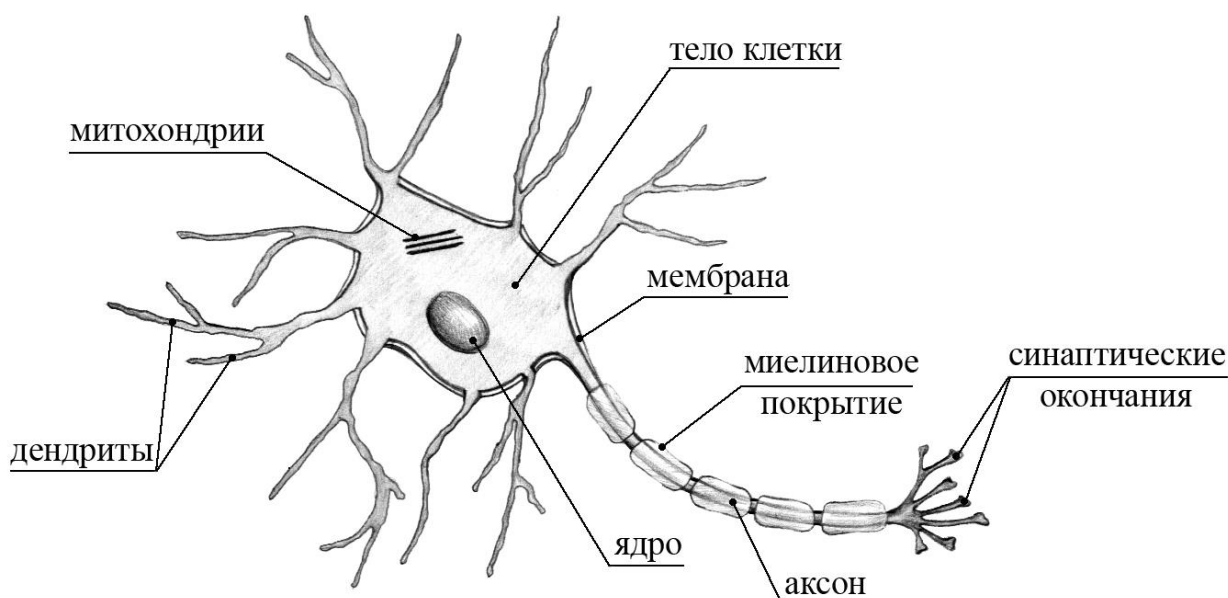


Рисунок 1 – Строение нейрона головного мозга

Мозг состоит из нейронов, каждый из которых имеет один длинный отросток, аксон, и множество коротких отростков, дендритов, которые связываются синапсами с аксонами других нейронов. Каждый нейрон периодически посылает по аксону нервный импульс. Таким образом, пока нейроны продолжают подавать сигналы, они являются живыми и могут находиться в выключенном состоянии, когда частота подачи сигналов мала, и в включенном состоянии, когда частота подачи импульсов увеличивается. Активация нейронов зависит от сигналов, проходящих через синапсы и дендриты других нейронов.

Искусственные нейронные сети – это машинная интерпретация мозга человека, в котором находятся нейроны, передающие информацию в виде электрических импульсов.

Связь между нейробиологией и нейронными сетями очевидна. В обоих случаях присутствуют нейроны, имеющие связи, по которым с помощью импульсов передается информация.

Но существуют и различия. В головном мозге нейрон подает сигналы через случайные промежутки времени, а в компьютере импульс передается с определённой точностью. Таким образом, в некоторых случаях искусственные нейронные сети превосходят работу человеческого мозга по скорости, то есть можно сказать, что нейронные сети могут быстрее решить поставленную задачу, оптимизируя ее, тем самым помогая получить результат за меньшее количество времени.

1.2 Классификация и строение нейронных сетей

Каждая нейронная сеть включает в себя первый слой нейронов, их основной задачей является принимать и распределять входные сигналы по остальным нейронам. Этот слой называется входным. Он является общим для всех типов нейронных сетей. Остальные слои и их количество различны, зависят от типов нейросетей.

Искусственные нейронные сети могут быть однослойными или многослойными.

Однослойная нейронная сеть — это структура, в которой взаимодействуют нейроны, где сигналы от входного слоя сразу передаются на выходной слой, который и дает результат.

Многослойная нейронная сеть помимо входного слоя и выходного слоя, имеет промежуточные слои, на которых осуществляется предварительная обработка и распределение информации. Количество промежуточных слоев показывает, насколько сложна нейронная сеть.

Так же нейросети классифицируются по

- 1) типу нейронов (однородные, гибридные)
- 2) методу обучения (с учителем, без учителя, с подкреплением)
- 3) типу входной информации (аналоговые, двоичные, образные)
- 4) характеру настройки синапсов (с фиксированными связями, с динамическими связями)

Искусственная нейронная сеть представляется в виде направленного графа, узлами которого являются нейроны, имеющие между собой связи.

Рассмотрим 3 класса нейронных сетей:

- 1) однослойные сети прямого распространения (рисунок 2.а)
- 2) многослойные сети прямого распространения (рисунок 2.б)
- 3) рекуррентные сети (рисунок 2.в)

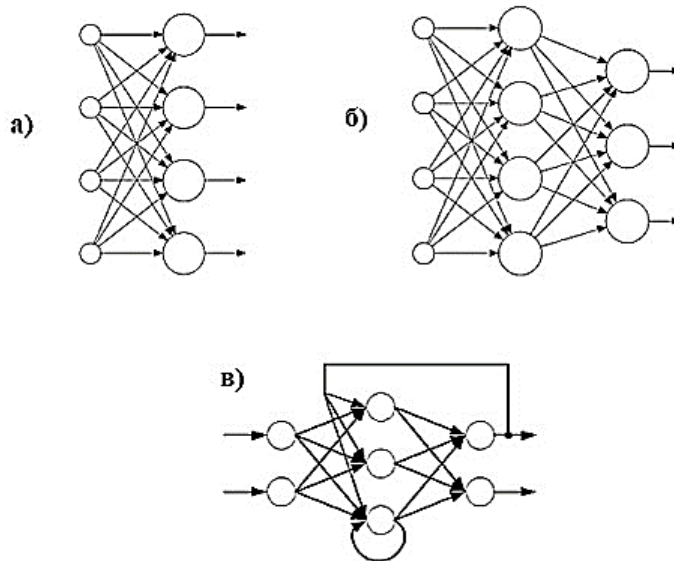


Рисунок 2 – Архитектура искусственных нейронных сетей

1.3 Принцип работы искусственных нейросетей

Принцип работы искусственной нейронной на рисунке 3 сети состоит в следующем:

- 1) На входной слой нейронов поступают определенные данные. Представим входные данные вектором вещественных чисел $x = (x_1, x_2, \dots, x_d) \in R^d$.
- 2) Каждый синапс, через который передаются данные следующему слою, имеет собственный коэффициент веса w_i . Состояние текущего нейрона вычисляется по формуле:

$$S = \sum_{i=1}^d x_i w_i \quad (1)$$

- 3) Для нормализации входных данных используется нелинейная функция активации, без которой невозможно будет составить содержательную сеть.

4) Подставляя S в формулу активации, происходит формирование выходных данных и передача их дальше до тех пор, пока не дойдут до конечного выхода.

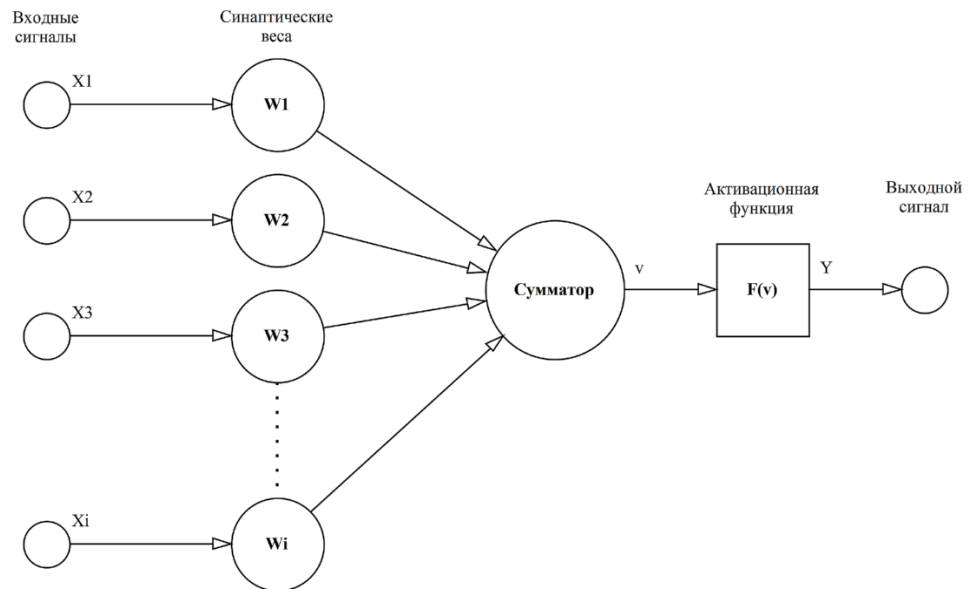


Рисунок 3 – Принцип работы нейронных сетей

Рассмотрим функции активации:

1) логистический сигмоид (рисунок 4):

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

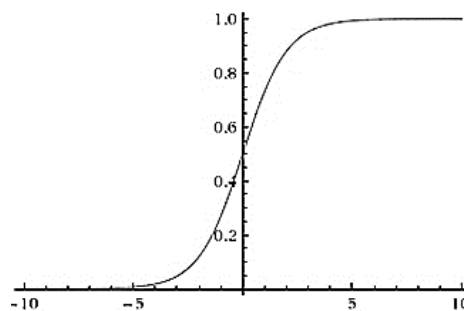


Рисунок 4 – Логистический сигмоид

- 2) гиперболический тангенс (рисунок 5):

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3)$$

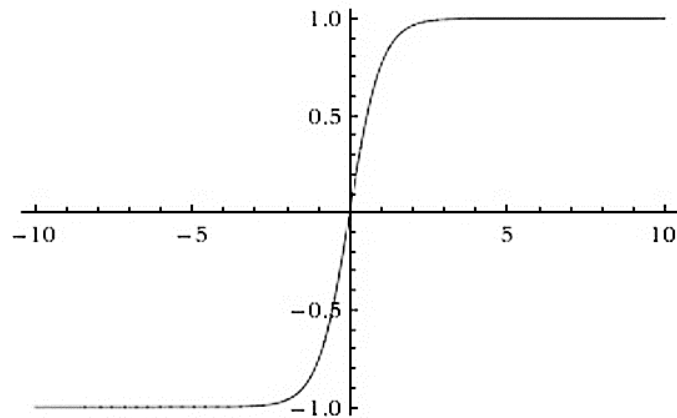


Рисунок 5 – Гиперболический тангенс

- 3) ступенчатая функция (Хевисайда) (рисунок 6):

$$\text{step}(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0 \end{cases} \quad (4)$$

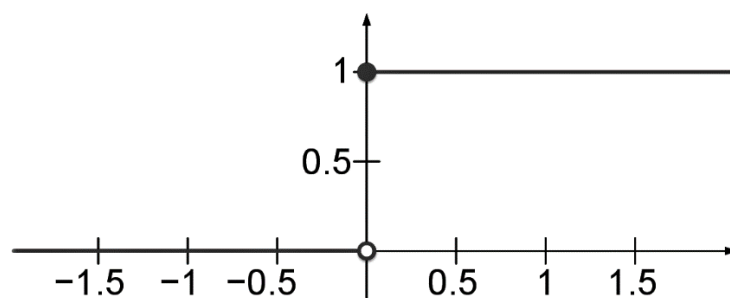


Рисунок 6 – ступенчатая функция

- 4) кусочно-линейная (рисунок 7):

$$\text{ReLU}(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (5)$$

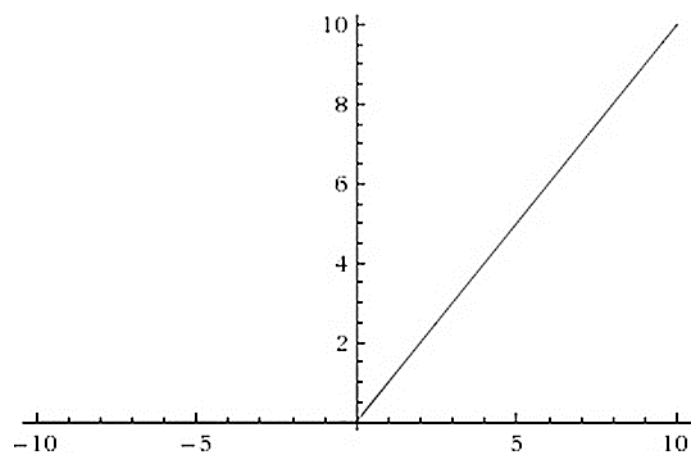


Рисунок 7 – кусочно-линейная функция

Существуют и другие функции активации, которые тоже показывают себя достойно в некоторых областях и экспериментах. Архитектура системы и алгоритм являются самой важной частью в построении искусственной нейронной сети.

2 Машинное обучение и работа с данными

2.1 Задачи машинного обучения

Машинное обучение – это некая модель, которая обучается определенным образом и выдает результат. По определению Томаса Митчелла в книге «Машинное обучение» [6]: «Компьютерная программа обучается по мере накопления опыта относительно некоторого класса задач T и целевой функции P , если качество решения этих задач (относительно P) улучшается с получением нового опыта». Выбор целевой определяет всю дальнейшую работу, которая приводит к получению модели.

Рассмотрим основные задачи машинного обучения, представленные на рисунке 8:

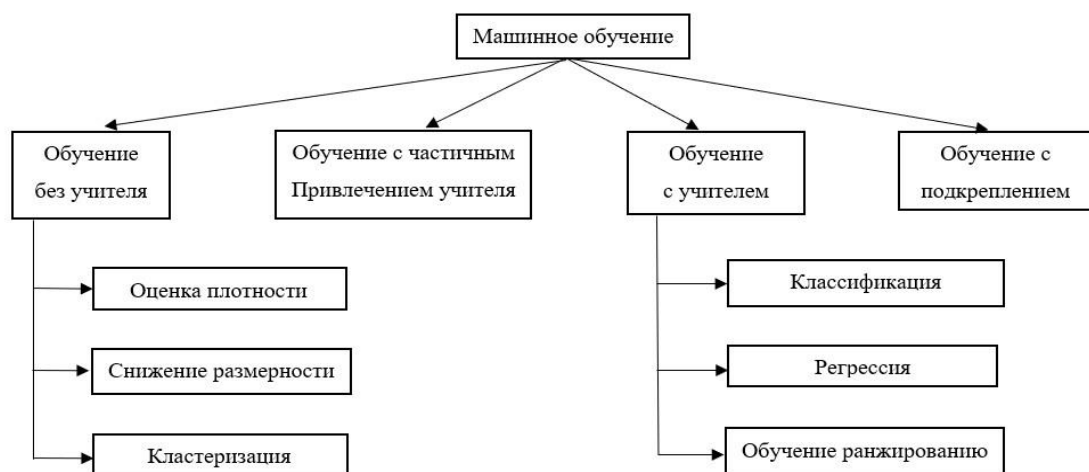


Рисунок 8 – Общая классификация постановок задач машинного обучения

1) Обучение с учителем

На основе тестового набора данных и известных ответов на новый опыт загружается новый обучающий набор данных, который сравнивается с известными ответами.

Задачи обучения с учителем делятся на задачи классификации и регрессии, а также на задачи обучения ранжированию.

В задаче классификации необходимо отнести входной объект к одному из классов. Например, по изображению животного определить кто это: кошка, собака, лошадь и так далее.

В задачах регрессии нужно предсказать значение функции, которая может иметь много разных значений. Например, предсказание курсов валют или предсказание завтрашней погоды.

Задача обучения ранжированию часто встречается в поисковых системах, где по имеющимся данным объекты ранжируются в порядке убывания целевой функции.

Обучение с учителем, таким образом, основано на больших объемах данных и готовых реакциях на тот или иной новый опыт.

2) Обучение без учителя

Если в задаче обучения с учителем результат основывается на готовых правильных ответах, то в задаче без учителя все иначе: для каждой конкретной задачи алгоритму не дается уже заранее подготовленный ответ.

Задания без учителя делятся на задачи кластеризации, уменьшения размерности и оценки плотности.

Задача кластеризации заключается в том, что нужно разбить данные на ранее неизвестные классы на основе некоторого сходства, чтобы объекты, отнесенные к одному классу, были максимально похожи друг на друга, а объекты, отнесенные к разным классам, были менее похожи. Например, сегментация медицинских изображений для обнаружения опухолей.

В задаче уменьшения размерности это означает, что необходимо установить представление данных меньшего размера из больших входных данных, чтобы свести к минимуму потерю исходных данных.

И, наконец, самый общий класс задач обучения без учителя – задача оценки плотности. Задача состоит в том, что необходимо оценить

распределение $p(x)$, из которого получаются представления о том, откуда взялись точки данных $\{x_1, x_2, \dots, x_N\}$.

3) Обучение с частичным привлечением учителя

Когда возникают трудности с выделением важных признаков из данных и маркировкой всех объектов, используется метод обучения с частичным участием учителя.

Обучение с помощью частичного привлечения учителя часто используется для решения медицинских задач, где небольшое количество размеченных данных может значительно повысить точность.

4) Обучение с подкреплением

Наиболее популярной средой тестирования для обучения с подкреплением является видеоигра. Агент пытается найти наилучший способ достижения цели или повышения производительности в конкретной среде. Когда агент совершает действия, помогающие достичь цели, он получает вознаграждение. Глобальная цель – предсказать следующий шаг, чтобы получить максимальную награду.

2.2 Работа с данными и классификация изображений

Как уже известно, построение нейронной сети основывается на большом количестве данных, с помощью которых сеть обучается. Точность результата зависит от набора данных: чем больше информации, по которой обучается нейросеть, тем выше точность конечного результата исследования, если информации недостаточно, то точность результата значительно снижается.

Определение количества наблюдений для обучения нейронной сети является сложной задачей. Для реальных задач достаточно не больше тысячи входных данных, а для более сложных проектов не будет хватать и миллиона.

Для хранения данных используют Microsoft Excel, JSON, базы данных. Так же данные могут быть смоделированы или симулированы, например, в MATLAB. В каком виде и где должны храниться данные – это отдельная

задача при практической реализации, так как формат данных уже зависит от конкретной задачи, решаемой нейронными сетями.

Входные данные не должны быть противоречивыми и должны содержать истинную информацию о предметной области, иначе это приведет к плохому качеству обучения сети и неверным результатам.

Задачи классификации изображений относятся к задачам ввода изображений в нейронную сеть, присвоения какой-либо метки для этого изображения или сравнения изображения с эталонным.

Для работы с изображениями наиболее эффективной является свёрточная нейронная сеть (CNN).

Сверточные сети работают не с данными в виде векторов, а с тензорами – массивами числовых матриц. Изображения представлены в компьютере пикселями, и каждый пиксель представляет собой значение интенсивности для соответствующего канала. Интенсивность каждого канала описывается целым числом от 0 до 255. Чаще всего используются цветные изображения, состоящие из пикселей RGB – пикселей, содержащих яркость трех каналов: красного, зеленого и синего. Различные комбинации этих цветов позволяют создать любой цвет по всему спектру.

Каждая тензорная матрица отвечает за интенсивность своего канала, а совокупность всех матриц описывает изображение целиком, поэтому наиболее эффективно для распознавания изображений использовать сверточные сети.

Чтобы выполнить распознавание или классификацию изображений, нейронная сеть должна выполнить извлечение признаков. Признаки – это данные, представляющие наибольший интерес, которые можно передать в нейронную сеть для анализа, чтобы классифицировать изображение в определенную категорию.

Признаки могут быть извлечены по-разному. Один из них – выделение признаков с помощью фильтров. Первый слой нейронной сети получает все пиксели изображения. После того, как все данные загружены в сеть, к изображению применяются различные фильтры, формирующие

представление о различных частях изображения. Процесс извлечения признаков из изображения выполняется с использованием свёрточных слоев, которые формируют представление части изображения. Размеры фильтра охватывают ширину и высоту, а глубина – это количество цветовых каналов, которые имеет изображение. Изображения в градациях серого имеют только 1 цветовой канал, тогда как цветные изображения имеют глубину 3 или 4 канала в зависимости от цветовой модели. Карта признаков является окончательным результатом всех вычислений.

Карта признаков передается в функцию активации, увеличивая ее нелинейность, а затем данные отправляются на слой пула (объединяющий слой), где данные сжимаются, а сеть становится более гибкой и способной лучше распознавать объекты и изображения.

Все слои должны быть плотно соединены друг с другом. Для этого значения сжимаются в длинный вектор или упорядоченный столбец чисел. Последние несколько слоев CNN образуют группы нейронов, которые представляют разные части рассматриваемого объекта. Когда достаточное количество этих нейронов срабатывает в ответ на входное изображение, оно классифицируется как объект. Нейроны среднего полносвязного слоя выводят бинарные значения, связанные с возможными классами. Нейрон имеет значение «1» для класса, представляющего изображение, и «0» для других классов.

2.3 Компьютерное зрение. Обработка изображений

В наше время все чаще появляются машины, техника, которая может помочь и минимизировать затраты времени на выполнение той или иной деятельности. Беспилотные автомобили, роботы, камеры, определяющие какое-либо происшествие и многое другое – все это не было бы возможным без компьютерного зрения.

Компьютерное зрение – это автоматический захват и обработка изображений неподвижных и движущихся объектов с помощью компьютерных средств. Сегодня компьютерное зрение широко используется во многих компонентах цифровой экономики: интеллектуальных транспортных системах, электронном здравоохранении, беспилотных автомобилях, дронах, высокотехнологичном сельском хозяйстве.

Обработка изображения является важным шагом для получения более точного результата нейросетью. Рассмотрим пример обработки изображения.

Для того, чтобы распознать определённый объект изображения необходимо скорректировать форму изображение, а именно выделить границу, где находится объект (сделать обрезку). Так же для улучшения качества изображения повышают резкость и накладывают фильтры. Чаще всего используют RGB модель, пример применения которой изображен на рисунке 9.

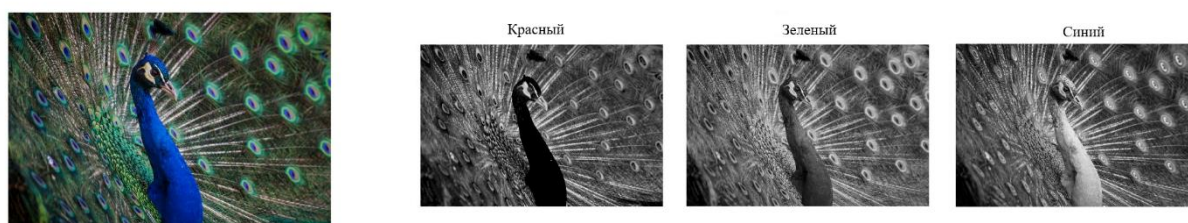


Рисунок 9 – Применение RGB модели

Это модель для отображения цвета в цифровых устройствах. Она по умолчанию используется при подготовке изображений для цифровых носителей. Различные комбинации цветов RGB модели позволяют создать любой из цветов всего спектра.

Помимо модели RGB так же существуют и другие цветовые модели:

- 1) Lab

Осями цветовой модели являются L – яркость объекта, a – градация от красного к зеленому, b – градация от желтого к синему. За единицу в модели принимается минимальное цветовое различие, воспринимаемое человеческим глазом. Поэтому Lab имеет максимальный цветовой охват.

2) HSB

Координаты данной модели: H – цветовой тон, S – насыщенность, V – яркость. Визуально цветовую модель HSB можно представить в виде цилиндра. Насыщенность и яркость варьируются от 0 до 100%, а тон измеряется в градусах от 0 до 360.

3) CMYK

Эта модель построена на смешении четырёх красок: C – сине-зеленый, M – пурпурный, Y – желтый, K – черный. Данная модель позволяет увидеть на электронных устройствах, как изменятся цвета на бумаге при смешении красок.

Вследствие поочередного выполнения этих действий получим изображение, на котором можно приступать обучать нейронную сеть.

2.4 Свёрточная нейронная сеть

Уже известно, что свёрточные нейронные сети хороши в распознавании изображений. Они состоят из блоков, которые накладываются друг на друга слой за слоем, получая более мощные архитектуры. Основными блоками этой сети являются свёрточные слои, слои пулинга (слои объединения), слой активации и полносвязные слои.

Свёрточные слои – это основные свёрточные блоки в CNN. На них приходится большая часть вычислительной нагрузки сети. Они выполняют скалярное произведение матриц, где одна матрица – это набор обучаемых параметров, также называемый ядром, а другая матрица – конечная часть рецептивного поля. Пример модели свёртки изображения представлен на рисунке 10.

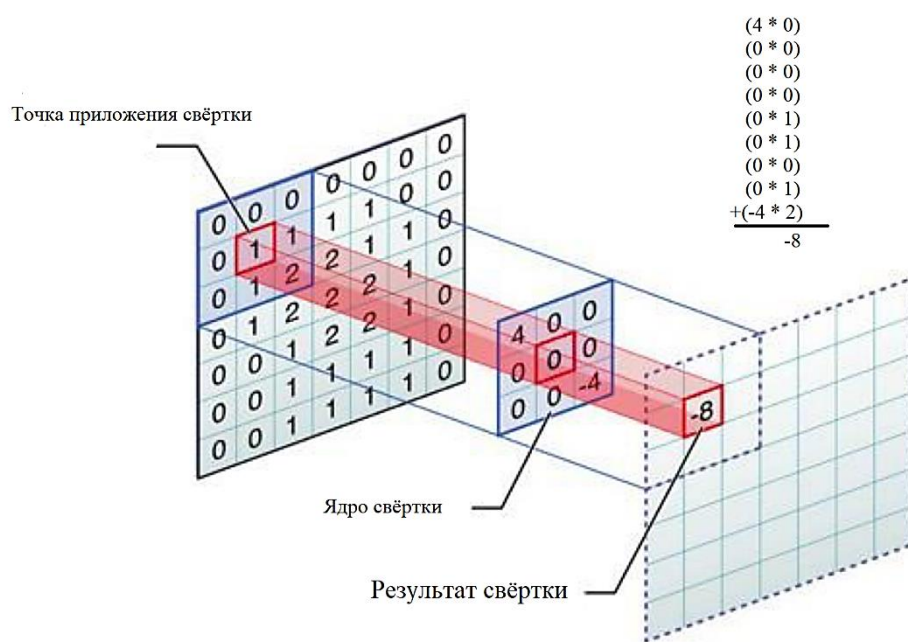


Рисунок 10 – Свёртка изображения

Слой пулинга заменяют выходные данные сети в некоторых местах, получая сводную статистику ближайших выходов. Это помогает уменьшить размер классификационного изображения, что, в свою очередь, уменьшает количество необходимых вычислений и весов. Операции объединения обрабатываются отдельно для каждого фрагмента представления.

Полносвязный слой – это слой в свёрточной сети, в котором нейроны полностью связаны со всеми нейронами в предыдущем и последующих слоях. Полностью связанные слои помогают сопоставлять представления между входными и выходными данными.

Слой свёрточной нейронной сети устроен так, чтобы сначала обнаруживать более простые узоры (линии, кривые), а затем более сложные узоры (лица, объекты). Используя CNN, можно обеспечить компьютерное зрение.

2.5 Преимущества и недостатки искусственных нейронных сетей

Современные нейронные сети еще далеки от «настоящего» искусственного интеллекта. По мнению Томаша Миколова есть два основных качества «настоящего» искусственного интеллекта: способность к коммуникациям для общения с людьми и получения информации об окружающем мире, способность к обучению и мотивации для этого обучения [7].

Все чаще в разных сферах деятельности нейросети показывают себя с лучшей стороны и завоевывают доверие.

К преимуществам искусственного интеллекта можно отнести:

- 1) Обработку большого количества данных. Программа может быстрее выдать результат какого-то исследования, основанного на большом количестве информации.
- 2) Адаптивное обучение. Сеть может обучаться различному поведению на основе входных данных, которые она получает. Обучение продолжается, и поведение сети меняется по мере поступления новых данных.
- 3) Отказоустойчивость. Нейронная сеть извлекает из потока данных важные атрибуты и дополняет их, слабо реагируя на новые искаженные свойства.

Также искусственные нейронные сети имеют и недостатки:

- 1) Неточность результатов. Нейронные сети не дают 100% точных результатов. Чтобы повысить точность результатов, потребуется огромное количество данных, опыт и знания людей, работающих в изучаемой области.
- 2) Многоступенчатое решение. Каждый нейрон независим и решает свою часть задачи, поэтому нейронная сеть не может решить задачу шаг за шагом, а для решения сложных задач необходимо использовать ансамбль сетей.

3 Анализ предметной области и метода решения задачи

3.1 Постановка задачи

Медицина является самой сложной, требующей много времени на получение результатов о состоянии здоровья человека, областью. Для оптимизации работы медицинского работника и для выявления за короткий промежуток времени каких-то отклонений, заболеваний, патологий рационально использовать нейросетевые технологии.

Внедрение нейронных сетей в медицину является актуальной задачей на сегодняшний день. Уже проделано множество исследований о применимости нейронных сетей в различных областях медицины, созданы модели для диагностики некоторых сердечных заболеваний, заболеваний легких и многих других, которые уже показали хорошие практические результаты и в скором времени, возможно, будут активно использоваться в медицинских учреждениях в качестве вспомогательного аппарата при диагностике [8].

В рамках выпускной квалификационной работы ставится задача сбора данных, их классификация и обработки изображений диаграмм ЭКГ с помощью языка программирования Python.

3.2 Построение математической модели

Для построения нейронной сети необходимо собрать данные, на которых будет обучаться сеть. Сотрудничая с врачом-кардиологом одной из краснодарский больниц, были собраны данные различных ЭКГ: диаграммы, их анализ и заключение врача.

Обработав полученные данные, была создана база данных, включающая в себя различные сердечно-сосудистые патологии, на ее основе будет происходить классификация и дальнейшее исследование.

В этом наборе содержится информация о ЭКГ 145 пациентов, каждое изображение кардиограммы имеет свою расшифровку - диагноз.

С помощью программы STATISTICA 10 проведем корреляционный анализ данных и построим сеть классификации по ключевым предикторам.

Из-за большого объема данных для обучения необходимо определить самую значимую для каждого выходного параметра информацию, которая больше всего влияет на получение того или иного результата. Для этого проведем корреляционный анализ и составим корреляционную таблицу (рисунок 11).

Корреляции (Sheet1 в Курсовая Данные) Отмеченные корреляции значимы на уровне $p < 0,05000$ N=149 (Построчное удаление ПД)			
Переменная	Тахикардия	АВ блокада	Нормальная ЭКГ
Возраст	-0,189003	0,368205	-0,235990
Синусовый ритм	-0,450720	-0,068743	0,230650
Синусовая аритмия	-0,029121	-0,027937	0,245683
Желудочковые экстрасистолы	-0,075198	-0,056893	0,028044
Предсердные экстрасистолы	-0,041323	-0,039643	-0,039026
Суправентрикулярные экстрасистолы	-0,050783	-0,048718	-0,047960
Нормальная ЭОС	-0,100644	-0,290173	0,175616
Нормальные комплексы QRS	-0,012551	-0,182008	0,029047
Артефакты в отведениях	-0,029121	-0,027937	-0,027502
QS-паттерн	0,031143	-0,027937	-0,027502
M-паттерн	-0,029121	-0,027937	-0,027502
RSR-паттерн	-0,078658	0,013774	-0,074284
Удлинение интервала QT	-0,041323	-0,039643	-0,039026
Нормальный интервал PQ	-0,053119	0,074543	-0,111940
Удлинение интервала PQ	-0,050783	0,220134	-0,047960
Укороченный интервал PQ	-0,036030	-0,091159	-0,089740
Ступенька перед комплексами QRS - дельта вол	0,024759	-0,080955	-0,079695
Неконкордантные комплексы QRS	0,172467	-0,039643	-0,039026
Регулярные комплексы QRS	0,001435	-0,039643	-0,039026
Нерегулярные комплексы QRS	0,154254	-0,056448	-0,055570
Широкие комплексы QRS	0,322961	0,047530	-0,164476
Высокие комплексы QRS	-0,029121	-0,027937	-0,027502
Узкие комплексы QRS	0,077607	-0,069617	-0,068533
Нормальные зубцы T	0,104281	-0,001017	0,148225
Нормальные сегменты ST	0,051945	-0,056893	0,172134
Горизонтальная депрессия сегмента ST в отведе	0,096062	-0,086171	-0,084830

Рисунок 11 – Корреляционная таблица данных

Красным цветом выделены те значения, которые являются статистически значимыми, они будут использованы нами для построения информационной и математической модели. Корреляционный анализ используется для установления статических связей между параметрами

оптимизации. Теснота связей количественно выражается величиной коэффициентов корреляции, которые вычисляются по формуле:

$$r_{xy} = \frac{\sum_{i=1}^N (x_i - X)(y_i - Y)}{\sqrt{\sum_{i=1}^N (x_i - X)^2} \sqrt{\sum_{i=1}^N (y_i - Y)^2}} \quad (6)$$

X, Y – средние значения переменных.

Связь существует, если $|r_{xy}| \sim 1$. Знак коэффициента корреляции отражает вид связи: положительная или отрицательная (возрастает или убывает у при росте x). Если изменение одной случайной величины не ведет к закономерному изменению другой случайной величины, но приводит к изменению другой статистической характеристики данной случайной величины, то подобная связь не считается корреляционной, хотя и является статистической.

Для проверки достоверности выбранных данных с корреляционной связью проводят оценку значимости линейного коэффициента корреляции. Оценка значимости коэффициента корреляции при малых объемах выборки выполняется с использованием t-критерия Стьюдента. При этом наблюдаемое (фактическое) значение этого критерия определяется по формуле:

$$t_{\text{набл}} = \sqrt{\frac{r_{y,x}^2}{1 - r_{y,x}^2}} (n - 2) \quad (7)$$

Критическое значение из таблицы значений t-критерия Стьюдента, в которой необходимо учитывать уровень значимости α и число степеней свободы $(n-2)$, сравнивается с вычисленным по формуле (7) значением.

Если $t_{\text{набл}} > t_{\text{табл}}$, то полученное значение коэффициента корреляции признается значимым (т.е. нулевая гипотеза, утверждающая равенство нулю коэффициента корреляции, отвергается). И делается вывод, что между исследуемыми переменными есть тесная статистическая взаимосвязь.

Корреляция между случайными величинами может быть положительной и отрицательной. При положительной корреляции возрастание одной случайной величины влечёт за собой в среднем возрастание другой. При отрицательной – возрастание одной случайной величины влечёт за собой в среднем убывание другой.

Для оценки тесноты связи между случайными величинами используют шкалу Чеддока (таблица 1):

Таблица 1 – Шкала Чеддока

Количественная мера тесноты связи	Качественная характеристика силы связи
0,1 - 0,3	Слабая
0,3 - 0,5	Умеренная
0,5 - 0,7	Заметная
0,7 - 0,9	Высокая
0,9 - 0,99	Весьма высокая

Приступим к построению нейронной сети. Построим нейронную сеть для АВ-блокады. Для построения модели было выбрано 4 значимых предиктора, все из которых являются количественными: нормальная ЭОС, нормальные комплексы QRS, удлинение интервала PQ, отклонение ЭОС влево.

Опираясь на них, обучим сеть. Методом “Автоматические нейронные сети” было построено 10 искусственных нейронных сетей, из которых выбрана сеть под номером 7, обладающая наилучшими прогностическими

способностями. На рисунке 12 представлено окно с построенной нейронной сетью.

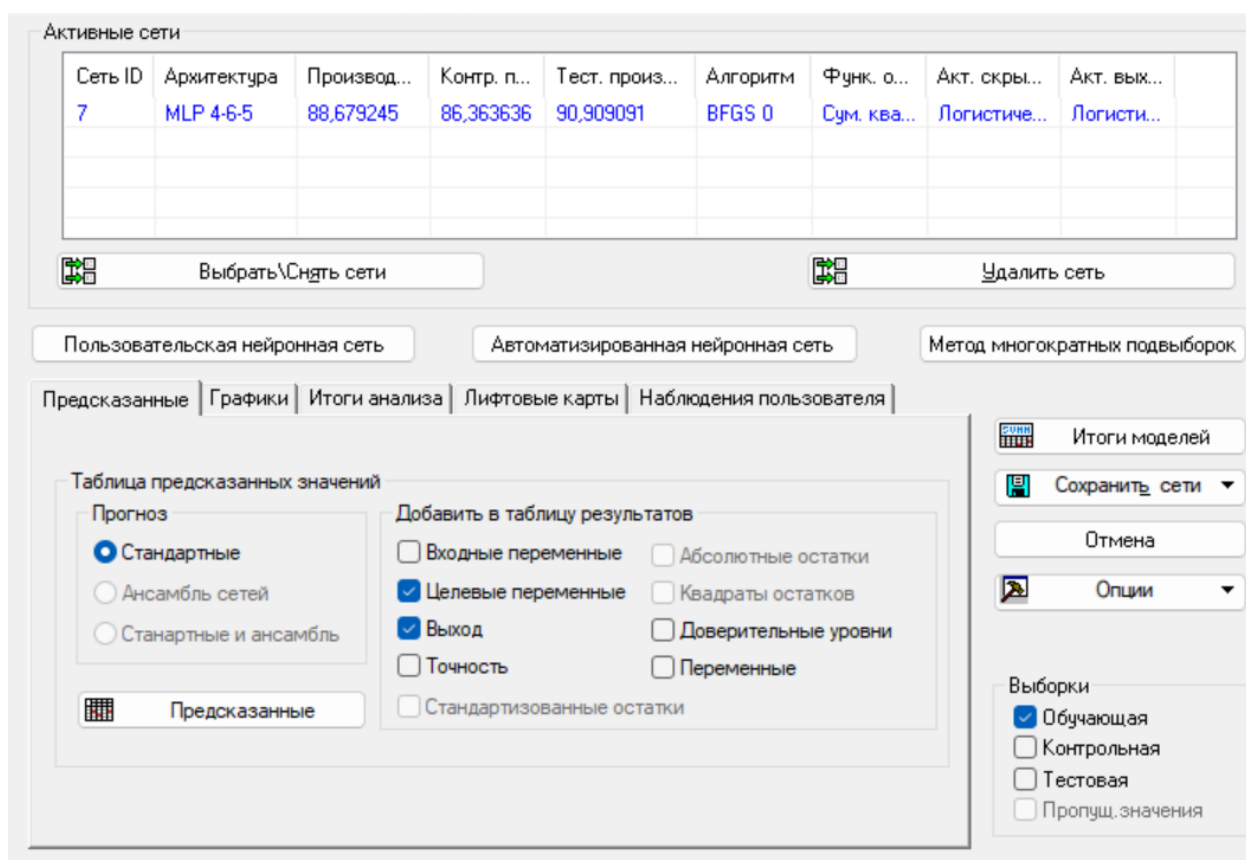


Рисунок 12 – Построение нейросети АВ-блокады

Тип нейронной сети – многослойный персептрон (MLP). Топология сети состоит из 4 входных нейронов, количество которых определяется как сумма количества непрерывных и значений качественных предикторов. Число скрытых нейронов – 6, а выходных нейронов – 5, определяющие разновидность АВ-блокады: АВ-блокада I степени, II степени, III степени и полная АВ-блокада, либо её отсутствие.

Случайным образом программой данные были разделены на обучающую (70%), тестовую (15%) и контрольную (15%) выборку. Обучающая выборка используется для обучения сети; тестовая – для тестирования ее производительности в процессе обучения, контрольная – для завершающей проверки, чтобы определить, насколько хорошо сеть прогнозирует «новые»

данные, которые не применялись для обучения. В столбцах 3, 4, 5 приведены производительности сети, являющиеся долей правильно классифицированных сетью больных, для обучающей, тестовой и контрольной выборки. Производительность обучающей выборки составила 88,7%, контрольной – около 86,2%, тестовой – 91%. Для обучения нейронной сети был использован итерационный квазиньютоновский метод численной оптимизации – алгоритм BFGS. Число 0 рядом с наименованием алгоритма обучения указывает на количество итераций, за которые сеть была обучена. В столбцах 7, 8, 9 указаны функции ошибки – сумма квадратов; активации скрытых нейронов – логистическая; а также функция выходных нейронов так же является логистической.

Для лучшего понимания насколько хорошо сеть обучилась рассмотрим матрицу ошибок, изображённую на рисунке 13, в которой из 106 значений 94 показали правильный результат и 12 – неправильный. Результат АВ блокада – 3 и АВ блокада – 4 не показали верных результатов из-за отсутствия достаточного количества данных.

Заключение: АВ блокада (Итоги классификации) (Корреляции для тахикардии в АВ-блокада.stw) Выборки: Обучающая		Заклучение: АВ блокада-0	Заклучение: АВ блокада-1	Заклучение: АВ блокада-2	Заклучение: АВ блокада-3	Заклучение: АВ блокада-4	Заклучение: АВ блокада-Все
7.MLP 4-6-5	Все	92,0000	5,00000	6,00000	1,0000	2,0000	106,0000
	Правильно	92,0000	1,00000	1,00000	0,0000	0,0000	94,0000
	Неправильно	0,0000	4,00000	5,00000	1,0000	2,0000	12,0000
	Правильно (%)	100,0000	20,00000	16,66667	0,0000	0,0000	88,6792
	Неправильно (%)	0,0000	80,00000	83,33333	100,0000	100,0000	11,3208

Рисунок 13 – Итоги классификации

Построим сеть для определения типа тахикардии у пациента. Для построения данной модели на основе корреляционного анализа было выбрано 10 значимых параметров, помещенных в таблицу 2, все из которых количественные: возраст, синусовый ритм, неконкордантные комплексы QRS, широкие комплексы QRS, блокада правой ножки пучка Гиса, отклонение ЭОС вправо, снижение сегмента ST в отведении V5, снижение сегмента ST в

отведении V6, депрессия сегмента ST в отведении V1, депрессия сегмента ST в отведении V4.

Таблица 2 – Коррелирующие значения с результатом тахикардии

Возраст	-0,118711
Синусовый ритм	-0,660970
Неконкордантные комплексы QRS	0,186162
Широкие комплексы QRS	0,460657
Блокада правой ножки пучка Гиса	0,448711
Отклонение ЭОС вправо	0,313354
Снижение сегмента ST в отведении V5	0,216927
Снижение сегмента ST в отведении V6	0,216927
Депрессия сегмента ST в отведении V1	0,209566
Депрессия сегмента ST в отведении V4	0,188898

В программе STATISTICA 10 на основе выбранных предикторов было построено 35 искусственных нейронных сетей, из которых отобрали сеть, обладающую наилучшими прогностическими способностями.

Построенная модель имеет архитектуру MLP 10-8-7, где MLP – многослойный персептрон. Описание построенной модели сети помещено в таблицу 3.

Таблица 3 – Модель сети, прогнозирующей тип тахикардии

Архитектура	Производительность обуч.	Контр. производительность.	Тест. производительность.	Алгоритм обучения	Функция ошибки	Ф-я актив. скрытых нейр.	Ф-я актив. выходных нейр.
MLP 10-8-7	97,80220	89,47368	89,47368	BFGS 93	Сум. квадр.	Гиперболическая	Экспонента

Топология сети состоит из 10 входных нейронов, количество которых определяется как сумма количества непрерывных и значений качественных предикторов. Число скрытых нейронов – 8, а выходных нейронов – 7, определяющие разновидность тахикардии: желудочковая, синусовая,

предсердная, с широкими комплексами, суправентрикулярная, тахикардия из АВ соединения, либо её отсутствие. На рисунке 14 изображена графическая модель построенной сети.

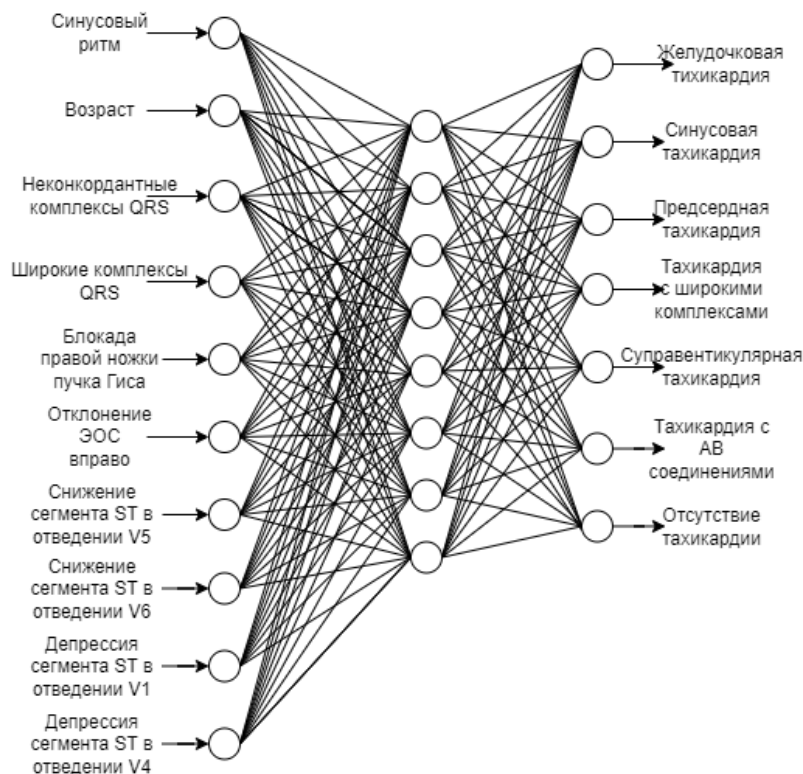


Рисунок 14 – Модель нейронной сети

Производительность обучающей выборки составила 97,8%, контрольной – около 89,5%, тестовой – 89,5%. Общая прогностическая способность сети 92,3%. Для обучения нейронной сети был использован итерационный квазиньютоновский метод численной оптимизации – алгоритм BFGS. Число 93 рядом с наименованием алгоритма обучения указывает на количество итераций, за которые сеть была обучена. В столбцах 7, 8, 9 указаны функции ошибки – сумма квадратов; активации скрытых нейронов – гиперболическая; функция выходных нейронов – экспонента.

Для лучшего понимания насколько хорошо сеть обучилась рассмотрим матрицу ошибок, помещенную в таблицу 4, в которой из 91 значения 89 показали правильный результат и 2 – неправильный.

Таблица 4 – Матрица ошибок

	Нет тахикардии	Желудочковая тахикардия	Синусовая тахикардия	Предсердная тахикардия	Тахикардия с широкими комплексами	Суправентрикулярная тахикардия	Тахикардия из АВ соединения	Тахикардия -Все
Все	77	5	3	1	2	2	1	91
Правильно	77	5	2	1	1	2	1	89
Неправильно	0	0	1	0	1	0	0	2
Правильно (%)	100,00 0	100,000	66,667	100,00 0	50,000	100,000	100,000	97,80 2
Неправильно (%)	0,000	0,000	33,333	0,000	50,000	0,000	0,000	2,198

Заметим, что двоих больных с синусовой тахикардией сеть прогнозировала верно, а у одного пациента синусовая тахикардия предсказана сетью как предсердная, таким образом отношение к пациенту будет более внимательным. Так же сеть предсказала тахикардию с широкими комплексами, спрогнозировав ошибочно диагноз одного пациента, определив вместо правильного диагноза желудочковую тахикардию.

Сеть достаточно хорошо обучена.

С помощью гистограммы, изображенной на рисунке 15, покажем количественное распределение различных типов тахикардии у пациентов.

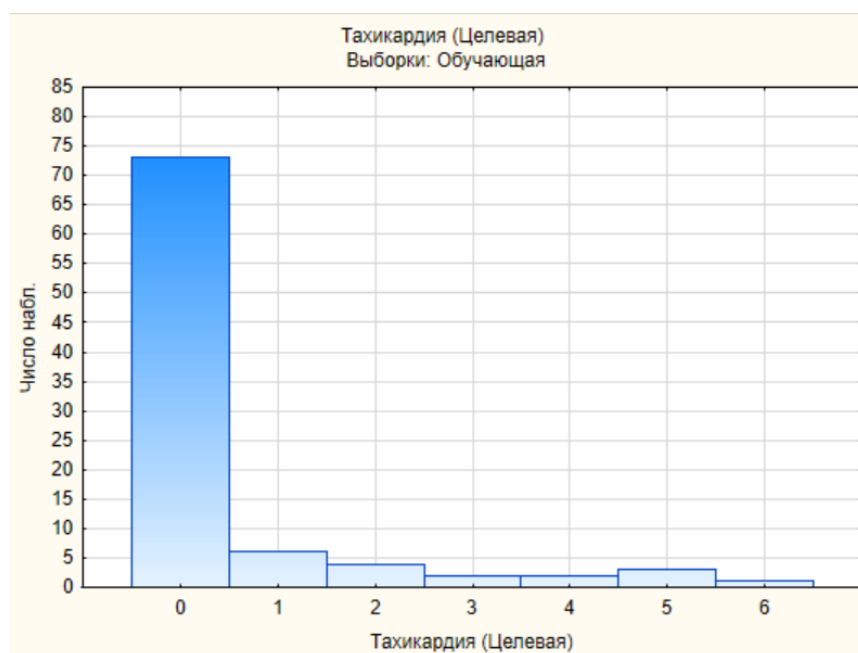


Рисунок 15 – График количества наблюдений видов тахикардий

Можно отметить, что пациентов с отсутствием патологии – 0, значительно больше, чем больных с различными степенями тахикардии: 1 – желудочковая, 2 – синусовая, 3 – предсердная, 4 – с широкими комплексами, 5 – суправентрикулярная, 6 – тахикардия из АВ соединения. Поэтому предложенную нейронной сетью классификацию будем считать приемлемой и использовать ее для написания приложения.

3.3 Обработка и классификация изображений с помощью Python

Обработка изображений является важной частью для их классификации и построения свёрточной нейросети.

Нейросетевые технологии быстро развиваются в сферах, где с помощью камеры, фотографии, изображения необходимо по определенным признакам провести классификацию. Например, 15 октября 2021 года на всех станциях московского метро заработал Face Pay – сервис оплаты проезда с помощью распознавания лиц.

У системы распознавания лиц две ключевые задачи: транспортная безопасность и удобство для пассажиров. С транспортной безопасностью понятно – система помогает правоохранительным органам ловить подозреваемых в преступлениях. А удобство пассажиров – это оплата с помощью распознавания лиц. Чем больше способов оплаты проезда, тем меньше очередей.

Нейросеть разбивает картинку на миллион точек, строит математическую модель и биометрический вектор – совокупность этих точек. Грубо говоря, переводится 3D-изображение лица в плоскую 2D-картинку, из которой извлекаются точки для сравнения с биометрическим вектором из базы.

Этот набор зашифрованных символов хранится на серверах. Когда человек подходит к турникету, камера на нём делает несколько снимков. Лучший кадр по освещённости, положению головы, открытости лица и многим другим параметрам переводится в биометрический вектор, сравнивается с точками в базе.

Такая система, реализованная с помощью нейронных сетей, значительно сокращает время прохождения через турникет и обеспечивает безопасность в общественном месте [9].

Так же классификация изображений с помощью нейросетей все чаще и чаще используется в медицине. Они используются для выявления патологий на рентгеновских снимках, в стоматологии, в кардиологии.

Чаще всего у каждого человека возникают те или иные проблемы с сердцем. Чем раньше среагировать на неважное состояние и на боль в грудной клетке, тем меньше риск плохого исхода.

Внедрение нейронных сетей в кардиологию – это все больше спасенных людей из-за скорости получения результата. С помощью них расшифровку ЭКГ можно получить за считанные секунды, что представляет удобство и скорость для врача-кардиолога, а также для выявления патологии сердца самостоятельно пользователем.

Перед обучением нейросети необходимо произвести обработку изображений: обрезать, наложить фильтры, сделать больше контрастность и резкость, тем самым сеть будет лучше обучаться и выдавать более достоверный результат и правильно классифицировать.

Классификацию картинок удобнее всего делать с помощью языка программирования Python и встроенной в него библиотеки TensorFlow и Keras.

TensorFlow – это библиотека машинного обучения с открытым исходным кодом, разработанная Google для решения проблемы построения и обучения нейронных сетей для автоматического поиска и классификации образов. Она также используется для обработки изображений с использованием различных встроенных методов. Вычисления TensorFlow представлены как поток данных через граф состояний. TensorFlow получил свое название от операций над массивами многомерных данных, также известных как «тензоры».

Keras – это библиотека глубокого обучения с высокоуровневым API, написанным на Python и работающим поверх TensorFlow. Keras позволяет легко и быстро создавать прототипы, поддерживая как сверточные, так и рекуррентные сети и их комбинации, и без проблем работает на процессорах (CPU) и (GPU). Базовая структура данных библиотеки Keras представляет собой модель, описывающую организацию слоев.

ЭКГ пациентов представляют собой картинку с расширением (.jpg). С помощью библиотек Python проведем обработку изображений кардиограммы. Результат обработки представлен на рисунке 16.

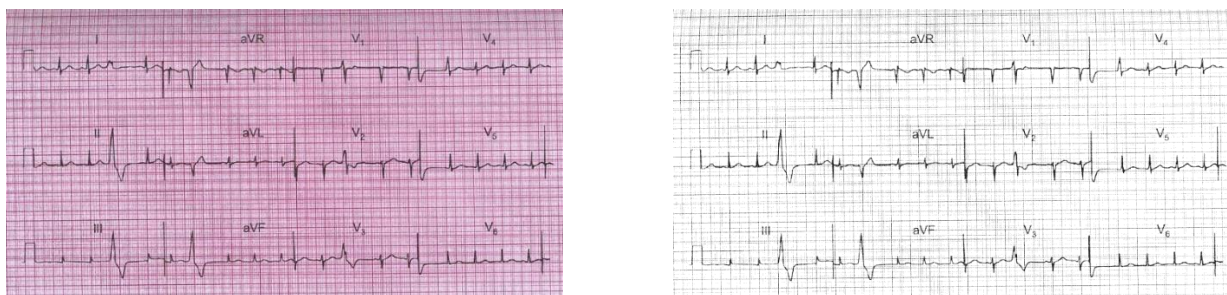


Рисунок 16 – Наложение фильтров на изображение

В итоге, стала лучше выделяться полоса сердцебиения во всех отведениях из-за применения черно-белого фильтра и усиления контраста, тем самым сети будет легче классифицировать изображение и выявить признаки.

Пакеты и библиотеки Python дают массу возможностей для работы с изображениями, их классификации и построение нейросети. Язык Python очень удобен для программирования из-за компактности и простоты.

4 Решение поставленной задачи

4.1 Назначение системы и ее пользователи

Как было сказано ранее, свёрточные сети – это мощная модель, представляющая собой будущее в распознавании образов и в компьютерном зрении в разных сферах деятельности человека.

Использование компьютерного зрения можно наблюдать уже в гипермаркетах. Например, с зимы 2023 года сеть розничных магазинов «Магнит» использует компьютерное зрение для распознавания товара на весах, что значительно ускоряет процесс взвешивания.

В наше время жизнь течет очень быстро, поэтому современному человеку необходимы как можно лучше оптимизированные процессы.

В данной работе рассматриваются свёрточные нейронные сети, с помощью которых по изображению ЭКГ прогнозируются патологии сердца у пациентов.

Система предназначена для различных пользователей: врачи, студенты медицинских учреждений и обычные люди, которые хотят отслеживать состояние здоровья.

Система будет иметь вид десктопного приложения с возможностями по загруженному изображению ЭКГ узнать о наличие/отсутствии заболевания сердца, вести динамику исследований, занося результаты в таблицу.

Таким образом, данное приложение поможет выявить наличие патологий и вести статистику исследований.

Для наиболее структурированного и схематичного описания возможностей системы используем диаграмму прецедентов UML, изображенную на рисунке 17.

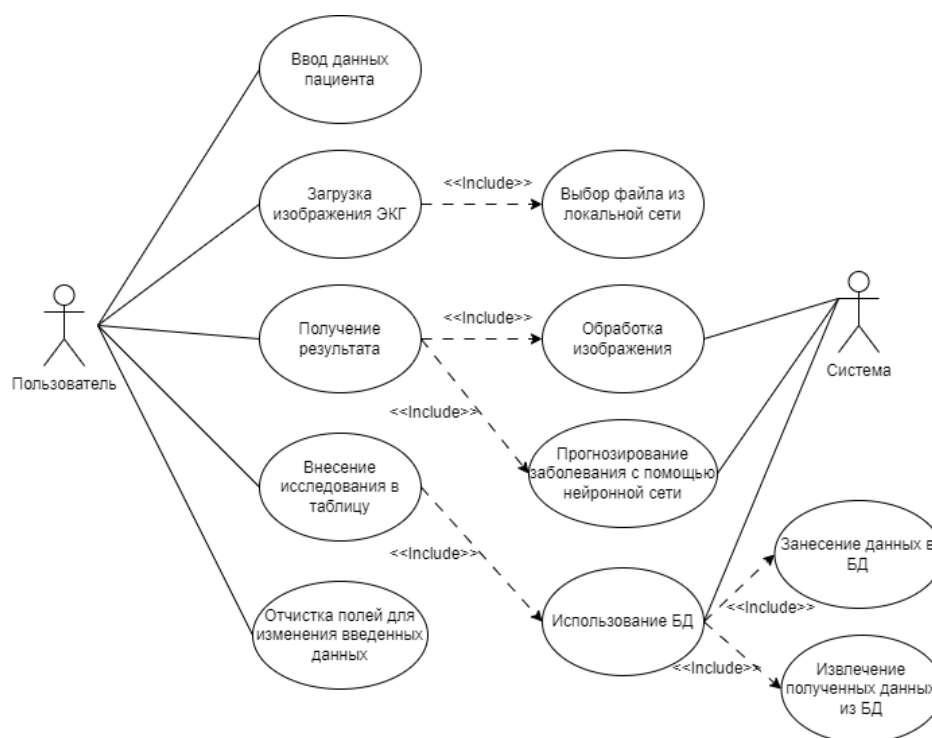


Рисунок 17 – Диаграмма прецедентов UML

UML – это набор правил, по которым нужно рисовать схемы. Зная его, можно быстро создавать универсальные графические представления сложных процессов и структур. Именно поэтому IT-специалисты во время разработки ПО часто используют UML-моделирование и проектирование процессов.

4.2 Описание построения нейронной сети

Построение свёрточной нейронной сети – нелегкий процесс, требующий внимательности и акцентирование на малейших деталях, начинающийся от выбора данных для обучения до наилучшего выбора количества слоев, функций активации и ошибки.

Для построения нейросети лучше использовать язык программирования Python.

Python – это язык программирования, широко используемый в интернет-приложениях, разработке программного обеспечения, науке о данных и машинном обучении (ML). Разработчики любят использовать Python, потому

что он эффективен, прост в освоении и кроссплатформенен. Программу с языком Python можно загрузить бесплатно, она совместима с различными системами и ускоряет разработку.

Для решения задачи данной работы на первом этапе построения нейронной сети необходимо разделить данные на обучающую (75%) и тестовую (25%). Всего используется 146 изображений ЭКГ, следовательно, обучающими будут 110 изображений, а тестовыми – 36.

На следующем этапе необходимо сопоставить картинки с соответствующими патологиями. В данной работе рассмотрим выявление различных видов тахикардии у пациентов.

Видов тахикардии 6: желудочковая, синусовая, предсердная, с широкими комплексами, суправентрикулярная, тахикардия из АВ соединения.

Каждому изображению соответствует вид тахикардии, либо её отсутствие.

Далее нужно выполнить обработку изображений.

Обучающие данные должны быть черно-белого цвета и иметь размер 256x256 пикселей.

С помощью встроенной библиотеки Pillow в Python произведем обработку.

После обработки изображений и сопоставления данных с картинкой можно приступить к построению свёрточной нейронной сети.

Проведем построение с помощью библиотек Keras, для классификации изображений и обучения сети, и Matplotlib, для построения графиков и анализа обучения сети.

Сеть содержит 6 слоев свёртки, 1 сглаживающий слой и 2 полносвязных слоя. В данной модели используется функция активации ReLU на свёрточных слоях и одном скрытом полносвязном слое. На выходном слое используется функция Softmax. В качестве метода оптимизации выбран метод Adam, являющийся модификацией градиентного спуска. Графическая модель свёрточной нейронной сети изображена на рисунке 18.

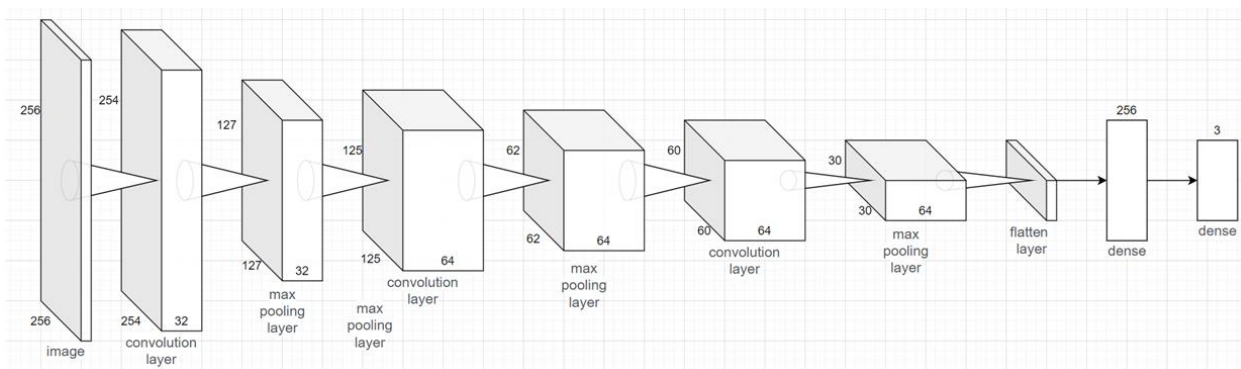


Рисунок 18 – Модель свёрточной нейронной сети

Первый сверточный слой принимает черно-белое изображение с размером 256×256 пикселей. Далее с помощью ядра свертки размером 3×3 вычисляются 32 фильтра, каждый из которых содержит сетку значений 254×254 – карту ответов фильтра, определяющую ответ этого шаблона фильтра для разных участков входных данных. В итоге получаем карту признаков с размером $254 \times 254 \times 32$, которая проходит через функцию активации ReLU для увеличения нелинейности значений. Выходная карта признаков меньше изначальной на 1 пискель с каждой стороны из-за эффекта границ.

Далее изображение подается на слой подвыборки, где из входной карты признаков с шагом 2 извлекаются выборки с размером 2×2 пикселя, откуда выбираются максимальные значения для каждого канала. На выходе получаем уменьшенную в 2 раза карту признаков размером $127 \times 127 \times 32$. Что позволяет сети выделять более сложные и обобщенные представления.

В конце данные проходят через полносвязные слои, которые анализируют входные признаки и объединяют их в различные атрибуты, которые помогут в классификации.

С помощью такой модели сеть была обучена за 100 эпох и достигла точности в 84% и 83% на обучающей и проверочной выборках соответственно. График изменения точности обучения в зависимости от эпохи изображен на рисунке 19.

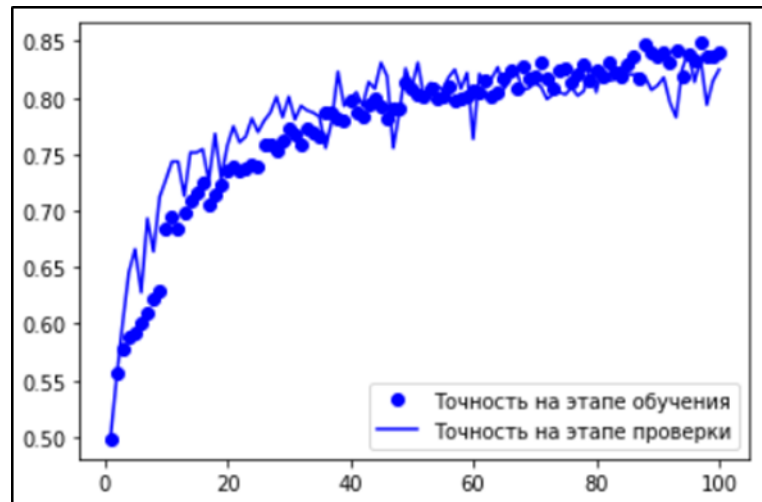


Рисунок 19 – График изменения точности обучения сети при 100 эпохах

4.3 Создание приложения

Для удобства использования предсказания кардиологического заболевания по изображению ЭКГ необходимо реализовать приложение с понятным интерфейсом и с возможностью использования необходимого функционала.

Десктопные приложения пишут на разных языках программирования: C++, C#, C, Python и других. Начинающим разработчикам проще всего использовать Python и его библиотеки для работы над графическими интерфейсами.

Одна из таких библиотек – Tkinter. Она входит в стандартный пакет Python и позволяет создавать приложения для Windows, macOS и Linux.

GUI (Graphical User Interface) – это графический пользовательский интерфейс, с которым мы взаимодействуем с помощью клавиатуры и мыши. В современных операционных системах почти все программы используют графический интерфейс и с GUI мы сталкиваемся каждый день: читаем новости или интересные статьи в браузере, используем различные приложения, мессенджеры.

Противоположностью GUI является командная строка, позволяющая управлять приложениями с помощью текстовых команд. Такой интерфейс реализован в терминале macOS и командной строке Windows.

В данной работе разработаем десктопное приложение на языке программирования Python с использованием библиотеки Tkinter.

Tkinter – это удобный интерфейс для работы со средствами Tk. Приложения, созданные с помощью Tkinter, могут запускаться на разных операционных системах, то есть являются кроссплатформенными приложениями.

Схематично работу с Tkinter можно представить в виде четырёх шагов, изображенных на рисунке 20.

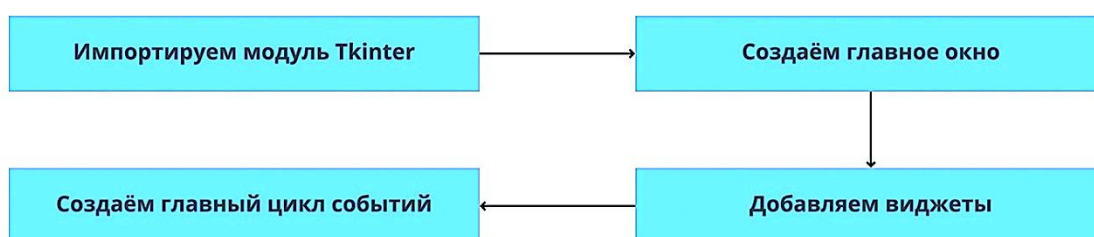


Рисунок 20 – Работа с библиотекой Tkinter

Алгоритм работы с библиотекой Tkinter:

- 1) Подключаем библиотеку Tkinter с помощью директивы `import`.
- 2) Создаём главное окно приложения, в котором будут размещаться все графические элементы.
- 3) Добавляем виджеты – визуальные элементы, выполняющие определённые действия.
- 4) Создаём главный цикл событий – он включает в себя все события, происходящие при взаимодействии пользователя с интерфейсом.

Ключевые объекты в работе с Tkinter – виджеты. Это аналоги тегов из HTML, которые позволяют создавать интерактивные и неинтерактивные элементы, например надписи или кнопки. Всего их 18, но чаще всего используют следующие:

- 1) Button – кнопки;
- 2) Menu – создание пунктов меню.
- 3) Canvas – «холст», на котором рисуют графические фигуры;
- 4) Label – контейнер для размещения текста или изображения;
- 5) Entry – поля ввода;
- 6) Radiobutton – радиокнопки для выбора подходящего варианта;
- 7) Checkbutton – флажки для выбора подходящего варианта;

Для создания приложения в решении задачи данной работы используются так же методы Tkinter:

- 1) ttk – предоставляет чуть больше функциональности по настройке и стилизации виджетов, чем tk.
- 2) messagebox – всплывающие окна сообщений.
- 3) filedialog – предоставляет функциональность файловых диалоговых окон, которые позволяют выбрать файл или каталог.

Дополнительно используются библиотеки Pillow с методами ImageTk, Image и tkcalendar с методом DateEntry.

4.4 Руководство пользователя

Приложение состоит из нескольких страниц, каждая из которых представляет некую информацию.

При запуске приложения необходимо авторизоваться. Окно авторизации представлено на рисунке 21.

Рисунок 21 – Окно авторизации

Авторизация представляет собой введение логина и пароля зарегистрированных пользователей. Если пользователь не зарегистрирован, то необходимо нажать на кнопку «Зарегистрироваться» и произвести ввод данных в окно регистрации, представленного на рисунке 22.

Рисунок 22 – Окно регистрации для новых пользователей

После удачно выполненной авторизации предусмотрено 2 исхода: работа с приложением для врачей-кардиологов и для обычных людей.

Для обычных пользователей предусмотрено 3 вкладки с различным функционалом.

Первая вкладка, изображенная на рисунке 23, – стартовая страница, на которой написано приветствие пользователя.

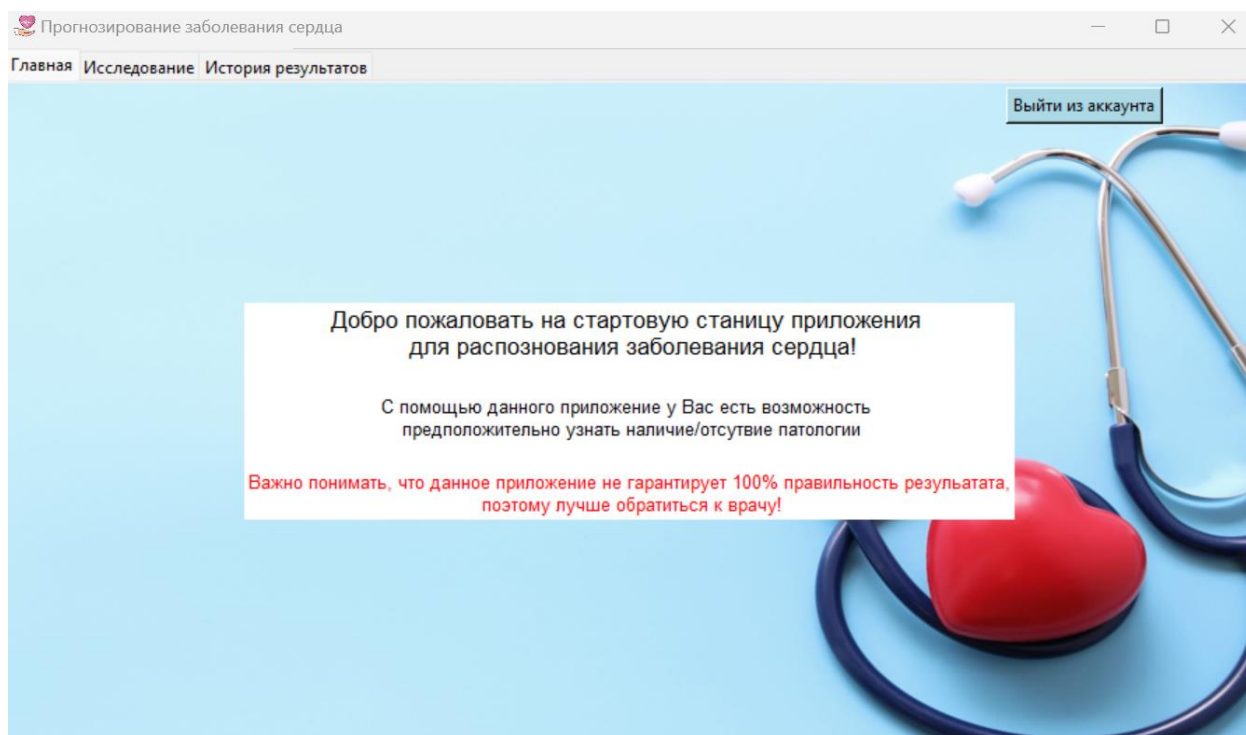


Рисунок 23 – Стартовая страница приложения для обычных пользователей

Вторая вкладка, изображенная на рисунке 24, – проведение исследования патологий сердца по изображению кардиограммы.

The screenshot shows a web application window titled "Прогнозирование заболевания сердца". It has three tabs: "Главная", "Исследование", and "История результатов". The "Исследование" tab is active. The main heading is "Введите свои данные для прогнозирования заболевания". Below this, there are three input fields: "ФИО" (a text box), "Дата рождения" (a dropdown menu showing "05-05-2023"), and "Дата исследования" (a dropdown menu showing "05-05-2023"). Below these fields, there is a label "Загрузите изображение ЭКГ" and a red button labeled "Загрузить изображение". In the center, there is a large empty rectangular box labeled "Место для изображения". Below this box is a green button labeled "Показать результат". At the bottom left, there is a label "Результат:". At the bottom right, there is a button labeled "Отчистить поля".

Рисунок 24 – Вторая вкладка приложения для обычных пользователей

На данной странице пользователю необходимо внести свои данные: фамилия имя отчество, дата рождения, дата исследования. Используя кнопку «Загрузить изображение» пользователю системы необходимо загрузить изображение ЭКГ с локальной сети для проведения исследования.

После того, как все данные введены и изображение загружено, после нажатия кнопки «Показать результат» программа начнет прогнозировать наличие либо отсутствие патологии.

После завершения исследования в строке «Результат» появится предполагаемое заболевание. Пример работы изображен на рисунке 25.

Прогнозирование заболевания сердца

Главная Исследование История результатов

Введите свои данные для прогнозирования заболевания

ФИО Слепцова Татьяна Павловна Дата рождения 21-10-2001 Дата исследования 05-05-2023

Загрузите изображение ЭКГ Загрузить изображение

Место для изображения

Показать результат

Результат: Тахикардии не обнаружено Отчистить поля

Рисунок 25 – Пример прогнозирования заболевания

Если данные введены некорректно, то нажав кнопку «Отчистить поля», все заполненные поля будут отчищены. Так же данную кнопку можно использовать, чтоб внести новые данные для проведения повторного исследования.

Третья вкладка, изображенная на рисунке 26, – история результатов.

На данной странице представлена таблица, в которой хранятся ранее проведенные исследования, что помогает отслеживать изменения и проводить статистику заболеваний пациентов.

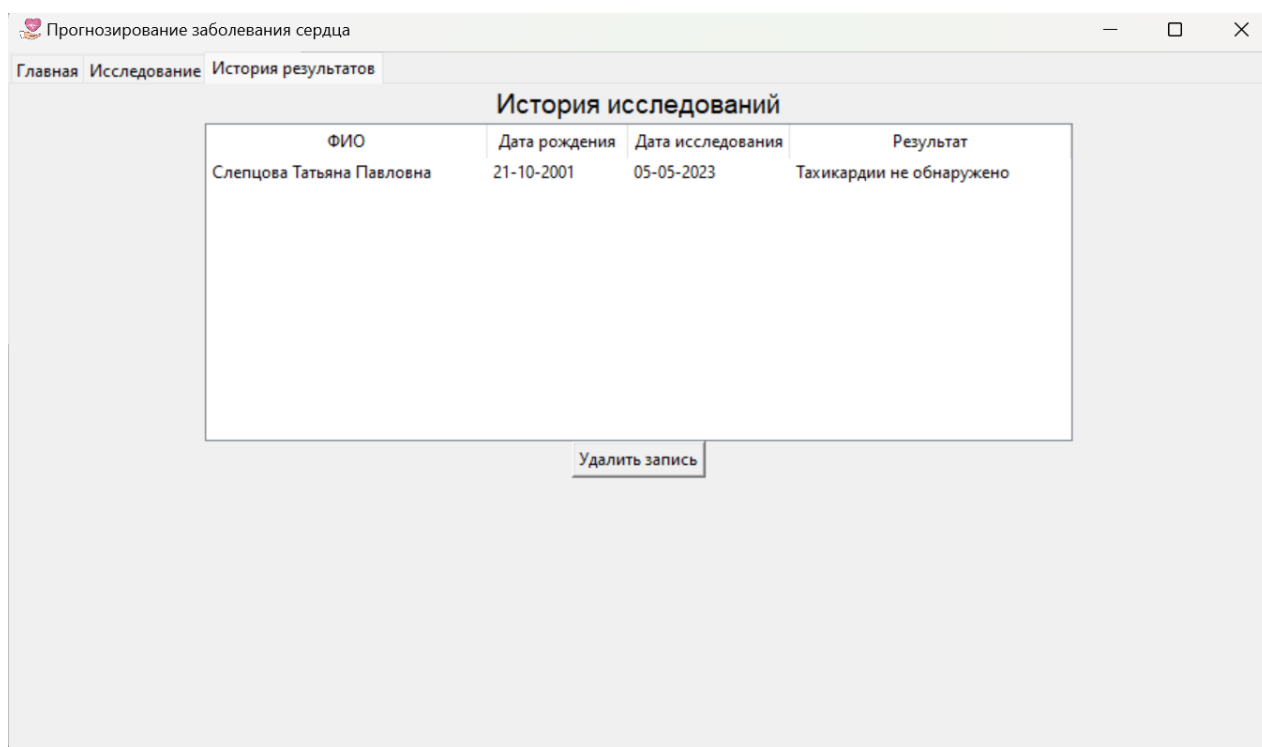


Рисунок 26 – Страница истории исследований для обычных пользователей

При обнаружении неверной записи, ее можно удалить, нажав кнопку «Удалить запись».

Для смены пользователя и выхода на окно повторной авторизации необходимо на главной странице приложения нажать на кнопку «Выйти из аккаунта», расположенную в верхнем правом углу. На рисунке 27 показано местонахождение кнопки.

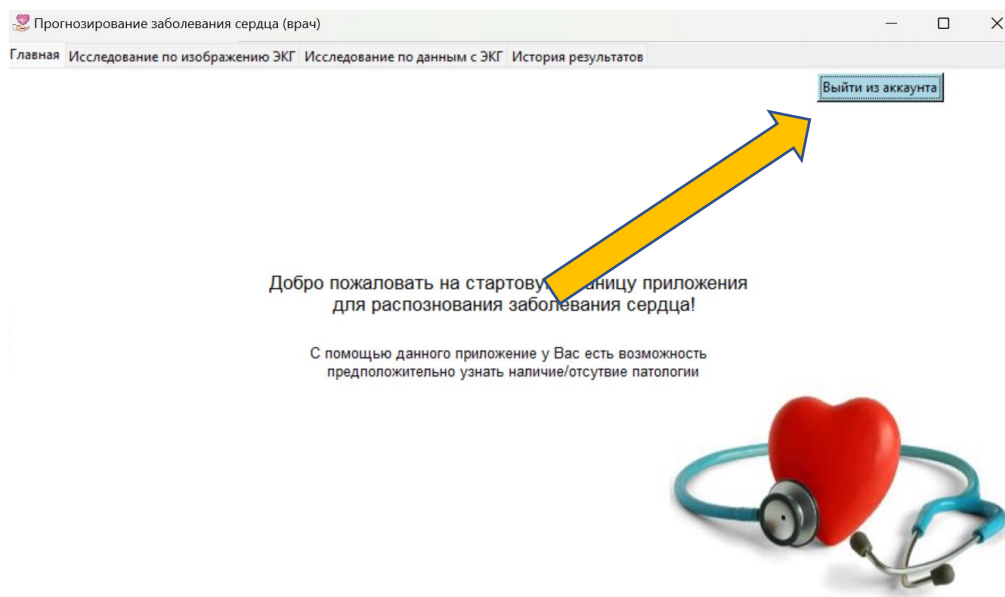


Рисунок 27 – Главная страница с указанием местонахождения кнопки «Выйти из аккаунта»

Для первичного входа в приложение для врачей необходимо ввести логин «doctor» и пароль «321». В дальнейшем есть возможность изменить эти данные.

Для врачей предусмотрен особый функционал, который позволяет прогнозировать различный вид тахикардии по данным с кардиограммы. Интерфейс окна представлен на рисунке 28.

Рисунок 28 – Дополнительное окно для прогнозирования тахикардии по данным с ЭКГ

После ввода данных необходимо нажать на кнопку «Показать результат». Каждое исследование заносится в таблицу, находящуюся на вкладке «История результатов».

Остальные вкладки с таким же функционалом, как у обычных пользователей.

Так же предусмотрен функционал выхода из приложения, интерфейс которого изображен на рисунке 29.

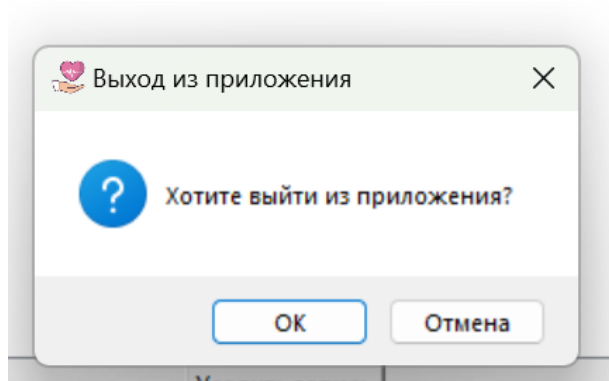


Рисунок 29 – Сообщение о выходе из приложения

Нажав крестик в верхнем правом углу приложения, появляется окно с подтверждением выхода.

Данный функционал ограждает пользователя от потери данных.

ЗАКЛЮЧЕНИЕ

Искусственные нейронные сети являются уже частью нашего мира. Они достаточно хорошо анализируют, могут распознавать изображения и относить их к определенному классу с помощью свёрточных сетей, которые достаточно часто применяются для решения прикладных задач в медицине и других областях. Как и любая система нейронная сеть имеет преимущества и недостатки при решении практических задач.

В рамках данной работы были изучены и проанализированы преимущества и недостатки программных средств для построения нейронных сетей классификации и свёрточных нейронных сетей, применение фильтров для входного изображения с помощью программных средств языка программирования Python: пакета Keras и библиотеки TensorFlow, сбор данных о пациентах с кардиологическими заболеваниями, и их анализ, определение связей между входной и выходной информацией.

Для пользователей написано десктопное приложение на Python, интерфейс которого реализован с помощью библиотеки Tkinter и её методов.

Приложение выполняет функцию прогнозирования типа тахикардии по изображению ЭКГ, а также по данным с ЭКГ. Предназначено ведение статистики исследований, путем занесения результатов наличия/отсутствия заболеваний в таблицу.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Обучение нейросети с учителем, без учителя, с подкреплением – в чем отличие? Какой алгоритм лучше? – URL: <https://neurohive.io/ru/osnovy-data-science/obuchenie-s-uchitelem-bez-uchitelja-s-podkrepleniem/> (дата обращения 10.05.2022)
2. Распознавание изображений на Python с помощью TensorFlow и Keras – URL: <https://evileg.com/ru/post/619/> (дата обращения 10.02.2023)
3. Халафян, А.А. Теория вероятностей и математическая статистика: учеб. Пособие / А.А. Халафян, Г.В. Калайдина, Е.Ю. Пелипенко. – Краснодар: Кубанский гос. ун-т, 2018. – 184 с. – ISBN 978-5-9518-0215-6.
4. Николенко, С. Глубокое обучение / С. Николенко, А. Кадури, Е. Архангельская. – СПб.: Питер, 2021. – 480 с. – ISBN 978-5-496-02536-2.
5. Haykin, S. Neural networks and learning machines / S.Haykin. – by Pearson, 2009. – 938 p. – ISBN 978-0-13-147139-9.
6. Mitchell, T.M. Machine Learning / T.M. Mitchell – New York: McGraw-Hill, 1997. – 432 p.
7. Micolov, T. A Roadmap Towards Machine Intelligence / T. Micolov, A. Joulin, M. Baroni – 2015. – URL – <https://arxiv.org/pdf/1511.08130.pdf> (дата обращения 22.12.2021)
8. Искусственный интеллект в медицине – URL: <https://www.kmis.ru/blog/iskusstvennyi-intellekt-v-mediticine> (дата обращения 15.05.2022)
9. Проект Face Pay – URL: [В метро Москвы теперь можно платить за проезд лицом. Спросили авторов проекта — как это работает — Промо на vc.ru](#) (дата обращения 15.05.2022)