**ВВЕДЕНИЕ**

В течение многих лет существовало мнение, что психически больные не хотели бы использовать технологии для лечения их состояния, в отличие от, например, людей с астмой или сердечной недостаточностью. Некоторые также утверждали, что лечение должно выполняться лицом к лицу, чтобы быть эффективным, и что технологии могут только испугать больного или ухудшить его паранойю.

Однако, результаты недавних исследований ряда престижных институтов, включая Гарвард, опровергают эти утверждения. Опыты показывают, что психически больные, даже те, кто тяжело болен, например, шизофренией, могут успешно контролировать свое состояние с помощью смартфонов, компьютеров и носимых датчиков. И эти инструменты только начало. В течение нескольких лет, новое поколение технологий обещает произвести революцию в практике психиатрии.

Когда пациент посещает психиатра или терапевта, большая часть времени первичного визита уходит на осмотр его симптомов, таких как сон, уровень энергии, аппетит и способность фокусироваться. Что также может быть очень сложным; депрессия, как множество других психических заболеваний, влияет на способность человека думать и запоминать.

Больной, скорее всего, уйдет со списком упражнений и рецептом лекарств. Есть вероятность, что медикаментозное лечение не будет эффективным, по крайней мере, в течение нескольких недель, план упражнений будет проигнорирован, что отрицательно отобразится на ход прогресса. К сожалению, врач ничего не узнает о пациенте до последующей встречи с ним.

Технологии могут положительно повлиять на сложившуюся ситуацию, предоставляя возможность в режиме реального времени следить за больным вне офиса. Вместо того чтобы полагаться только на упомянутые пациентом симптомы, доктор может взглянуть на данные о поведении больного с его персонального телефона и носимых датчиков. Психиатр может даже порекомендовать пациенту начать использовать такое устройство перед первым визитом.

Современные технологии и методы обработки данных, в частности машинное обучение, позволяют предоставить множество полезной информации из данных, которые, на первый взгляд, имеют мало общего со здоровьем человека.

Активное развитие технологий искусственного интеллекта, в настоящее время, привело к тому, что эти технологии широко применяются во многих областях. Компьютер как обычный вычислитель способен быстро и точно вычислять заданные параметры. Объединяя эти преимущества с моделями человеческого разума, технологии искусственного интеллекта улучшают человеческую жизнь, и даже, помогают выполнять задачи за пределами человеческих способностей. Искусственный интеллект – это комплексная и сложная наука, имеющая много пересечений с другими науками.

Актуальность магистерской диссертации заключается в использовании нейронных сетей для решения плохо формализованных задач в интеллектуальном анализе данных. Рост объемов информации, а также расширение круга технически сложных задач принятия решений требуют систематизации существующих методов и разработки новых методик и алгоритмов решения.

Искусственные нейронные сети дают многообещающие перспективы в развитии, а программное обеспечение имеет огромное преимущество от их использования. Кроме того, каждая реализуемая задача имеет неограниченный и нестандартный набор методов решения. В магистерской диссертации рассматривается возможность применения нейронной сети при решении задачи распознавания психологического состояния человека.

**1 ОБЗОР ЛИТЕРАТУРЫ**

* 1. **Машинное обучение**

Машинное обучение — это попытка машины найти закономерность между набором определённых свойств и заданным правильным ответом. Машина считается «обученной» тогда, когда она выводит из этого набора предметов и свойств закономерность. Эта закономерность называется моделью, а сам процесс вывода — тренировкой модели. Результат работы модели с данными напрямую зависит от того набора данных, который человек «скормил» модели при тренировке. Скорректировать такую модель в дальнейшем оказывается довольно сложно: этот процесс связан с долгой и затратной переподготовкой данных.

Из этого следует несколько выводов. Во-первых, машинное обучение очень плохо предсказывает или классифицирует то, что выходит за рамки модели

**1.2 Источники поведенческих данных**

Анализ больших данных позволяет получить точную информацию о человеке. Большие данные представляют собой огромный объем информации разного типа: картинки, видео, текст, геоданные, веб-журналы, машинный код. Вся информация находится в различных хранилищах и трудно поддается анализу с помощью традиционных методов. Для этого используются специализированные технологии, включая [искусственный интеллект](https://rb.ru/tag/ai/) и машинное обучение. Рассмотрим возможные источники данных, которые могут быть использованы для сбора поведенческой информации с целью определения психологического состояния человека:

GPS данные смартфона могут показать передвижения человека, которые, в свою очередь, отражают психическое здоровье человека. Посредством корреляции GPS измерений смартфона пациента и его симптомов депрессии, исследования 2016 года Центра Технологий Поведенческого Вмешательства в северо-западном Университете в Чикаго, было найдено, что когда люди в депрессии, они имеют тенденцию находиться дома больше чем, когда они чувствуют себя хорошо. Аналогично, люди входящие в маниакальную стадию биполярного расстройства, могут быть активнее и в движении. Консорциум по мониторингу, лечению и предсказанию стадий биполярного расстройства провел многочисленные исследования, которые демонстрируют, что этот тип данных может быть использован для предсказания течения биполярного расстройства [10].

Где GPS данные не доступны, могут быть использованы Wi-Fi и Bluetooth. Исследования Дрора Бен-Зеева, Университет Вашингтона, Сиэтл, показали, что Bluetooth-радиоприемники в смартфонах могут быть использованы для мониторинга места положения людей страдающих шизофренией в госпитале. Данные собранные через Wi-Fi сети могли бы также показать избегают ли пациенты, склонные к алкоголизму, бары и посещают ли встречи группы поддержки [11].

Данные акселерометра смартфона или фитнесс-трекера могут предоставлять более точные детали о передвижениях человека, определять тремор, который может быть побочным эффектом применения наркотиков, и захватывать шаблонные движения в упражнениях. Тестирование приложения названного CrossCheck недавно продемонстрировало как этот тип данных, в комбинации с другой информацией полученной с телефонов, может содействовать прогнозированию симптомов шизофрении, предоставляя информацию о сне и шаблонах поведения. Статья из Американского журнала психиатрии, Ипсит Вахия и Даниэль Сивел, описывает как у них получилось лечить пациентов с особо сложным случаем депрессии используя данные акселерометра. Пациент сообщал, что он был физически активен и проводил мало времени в постели, но данные с его фитнесс-трекера показали, что его воспоминания были ложны, доктора, таким образом, верно диагностировали его состояние как депрессию скорее чем, скажем, расстройство сна.

Отслеживание частоты телефонных звонков и текстовых сообщений может показать насколько человек общителен и указать на его психологические изменения. Когда одна из исследовательских групп Monarca взглянула на журнал входящих и исходящих текстовых сообщений и телефонных звонков, они пришли к выводу что изменения в этом журнале могли бы быть полезны для отслеживания депрессии также как и мании в биполярном расстройстве [12].

Физиологические данные, такие как пульс, гальваническая реакция кожи могут также пролить свет на умственное благополучие человека. Такого рода данные уже собираются некоторыми устройствами. Множество исследований показали, что изменчивость пульса может быть использована для отслеживания тяжести биполярного расстройства и шизофрении. Гальваническая реакция кожи, измерение электропроводимости кожи, которая зависит от состояния потовых желез и контролируется компонентом нервной системы, могут свидетельствовать о биологической активности, которая может быть сверхактивной при тревожном расстройстве.

В конечном счете, психиатрические расстройства это расстройства головного мозга, и использование электроэнцефалографии (ЭЭГ) для отслеживания активности головного мозга уже давно принятая практика в психиатрических исследованиях и лечении биологической обратной связи. Более новые, потребительские ЭЭГ гаджеты в наше время могут быть приобретены в магазине, но ещё не понятно насколько они эффективны [13].

Мобильный рынок предлагает пользователям смартфонов массу интересных приложений для поддержания здорового образа жизни и отслеживания собственных спортивных достижений. Сейчас подобные приложения могут измерять пульс или считать шаги, но делают это приблизительно и требуют от пользователей ввода определенных данных.

С помощью машинного обучения можно значительно улучшить показатели фитнес-приложений путем постоянного отслеживания физической активности человека без дополнительных действий с его стороны. То есть, пользователю не потребуется переключать режимы приложения, когда он переходит на бег или езду на велосипеде. Однако это можно реализовать при условии использования сенсоров, интегрированных с приложением.

Подобная задача решается в [приложении для людей, подверженных приступам мигрени, эпилепсии или обморокам](http://healint.com/). Приложение используется для мониторинга за состоянием пациентов и предупреждения критических ситуаций, отслеживает их нейробиологические сигналы через сенсоры смартфонов и обрабатывает информацию с помощью модели машинного обучения. Если возникает риск приступа, приложение «сигнализирует» об этом, отправляя сообщение близким, которые могут прийти на помощь [13].

**1.3 Распознавание эмоций**

Исследование методов автоматического распознавания эмоций позволяет дать компьютеру возможность оценивать настроение человека, для этого должны быть решены следующие задачи:

1) Захват и отслеживание лица. Задача захвата требует определения: присутствует ли лицо в поле зрения компьютера. Если присутствует, компьютер должен получить размер лица и его координаты положения на изображении. А отслеживание требует дальнейшего определения размера и положения, изменяющихся с течением времени.

2) Распознавание и описание особенности лица. Эта задача требует определения положения органов (глаза, нос, рот и т.д.) на лице, а также должны быть определены формы этих органов.

3) Классификация. Согласно информации об особенностях лица необходимо определить: какой вид эмоции присутствует на изображении? Затем необходимо определить информацию о настроении для дальнейшей разработки интеллектуального интерфейса.

Анализ выражения лица является непростым направлением исследований в области распознавания образов, в основном из-за сложности получения точных признаков лица и его выражений. Так как разница в движении характерных точек между различными выражениями лица не велика, например: открытый рот не означает, что это смех, это может быть плач или удивление. Методы, указанные ниже, развились из методов распознавания лица, но применялись и для распознавания эмоций.

С одной стороны, существующие методы распознавания используют три характерных признака для распознавания [2-6]: уровень серого, движение и частоты. Разные эмоции имеют приводят к разному уровню серого на цифровом изображении лица пользователя. Однако анализ этого характерного признака требует серьезной предварительной цифровой обработки изображения для нормализации входного изображения. Характерный признак: движение, использует информацию об изменении положения определённых точек на лице. Характерный признак: частоты, использует разницу между различными цифровыми изображениями лица в частотной области.

С другой стороны, методы распознавания можно разделить на три области [7-10]: целостные и локальные распознавания, эксрагирования деформации и движении, геометрические характеристики и характеристика внешности. Методы целостных распознаваний анализируют эмоции человеческого лица в целом, чтобы затем найти разницу между разными изображениями: Principal Component Analysis (PCA), Independent Component Analysis (ICA), Fisher’s Linear Discriminants (FLD), Local Feature Analysis (LFA), Fisher Actions, Hidden Markov Models, и кластерный анализ. Методы локальных распознаваний анализируют отдельные части лица, например: рот, брови и глаза. Типичные методы: Facial Actions Code System (FACS), Local PCA, Вейвлеты и Нейронные сети. Методы эксрагирования деформации и движений для распознавания берут за основу изменения органов на лице, когда меняются различные выражения: Active Shape Model (ASM), Point Distribution Model (PDM). Методы геометрических характеристиках основаны на форме и положении различных частей лица человека, чтобы извлечь характерные векторы, которые представляют геометрические характеристики лица. Однако рассмотренные методы не всегда позволяют распознать эмоции пользователя в реальном времени с достаточным уровнем точности на компьютеризированной платформе с небольшими вычислительными ресурсами (мобильные устройства, например), поэтому был разработан новый простой метод распознавания эмоций пользователя на лице человека.

Сегодня существует ряд готовых коммерческих решений и большое количество публикаций, посвящённых распознаванию эмоций [3–8]. В большинстве работ реализован подход, основанный на выделении признаков изображения: холистических (рассчитываемых в целом по всему изображению), локальных (вычисленных в окрестности заданных контрольных точек), темпоральных (определяемых по движению частей лица во времени) и последующей классификации этих признаков с помощью методов машинного обучения. Для выделения признаков лица исследователи в основном используют так называемые двигательные единицы лица (Action Units, AU), входящие в Систему кодирования лицевых движений (Facial Action Coding System, FACS), разработанную П. Экманом [3]. Базовые эмоции для классификации также выбираются согласно FACS: радость, злость, грусть, отвращение, страх, удивление (иногда в список базовых включают эмоцию презрения).

В работе [4] комбинируется несколько типов признаков (оптический поток, SIFT, иерархические гауссовы) с последующей классификацией методом опорных векторов (Support Vector Machine, SVM). Авторы публикации [5] значительных результатов достигли при использовании в качестве признаков пространственно-временной модификации локальных бинарных шаблонов (LBP-TOP [6]). В материалах [7, 8] продемонстрирован алгоритм расчета интенсивности AU и сопоставление эффективности различных групп признаков и их объединений. Интересный подход к классификации предложен в [9], где степени выраженности AU преобразуются в маркеры наличия эмоций с помощью логических деревьев решений, специфичных для различных этнических групп.

Существенным препятствием при разработке систем распознавания эмоций является ограниченность доступных баз данных, а также высокая доля индивидуальных особенностей в проявлении той или иной эмоции у разных людей. Это несколько ограничивает возможности применения машинного обучения для классификации, поскольку повышаются требования к обобщающей способности классификатора. Параметры изображения также существенно влияют на точность распознавания: изменение положения лица, наличие очков, макияжа или закрывающей брови причёски существенно затрудняют расчёт некоторых AU, а иногда и вовсе делают его невозможным. Поэтому для успешной работы системы необходимо учитывать эти факторы. Благодаря применению локальной фильтрации изображения при расчете степени выраженности AU, а также применению мультиклассификатора для оценки степени выраженности эмоций, можно снизить влияние этих помех.

Авторы предлагают систему распознавания базовых эмоций на основе алгоритма расчета степени выраженности AU ряда локальных признаков-маркеров по классификации П. Экмана. Полученные значения нормировали относительно нейтрального выражения лица того же субъекта, а затем рассчитывали степень выраженности каждой из базовых эмоций тремя классификаторами: нейронной сетью типа многослойного перцептрона (Multilayer Perceptron, MLP), вероятностным классификатором и системой логических правил.

**1.4 Распознавания эмоций из речи**

Важным каналом для идентификации эмоционального состояния человека является его речь. Различия в точности распознавания эмоций по голосу связаны в основном с модальностью эмоций. Любопытные данные были получены в отношении точности распознавания эмоций людьми разных профессий. Лучше всего распознавали эмоции врачи-психиатры по сравнению с инженерами, и даже по сравнению с врачами других специальностей. Исследование В. X. Манерова [10] идентификации эмоций по речи показало, что основным признаком, используемым человеком при слуховом восприятии эмоционально обусловленных изменений речи, является степень речевого возбуждения. Определение вида эмоции, переживаемой говорящим, осуществляется слушающим менее успешно, чем определение степени эмоционального возбуждения. Наиболее точно опознаются базовые эмоции, затем удивление и неуверенность и хуже всего — презрение и отвращение. На точность опознавания влияет способность диктора передавать в речи эмоциональные состояния. Выявлена тенденция лучшего распознавания положительных эмоций по сравнению с индифферентными и отрицательными эмоциональными состояниями.

Наиболее обширным исследованием в области распознавания эмоций является монография С. Стейдля [11], в которой рассмотрены разнообразные подходы к этой задаче. В частности, показано, что использование визуальных данных и транскрипции речи существенно повышает точность распознавания эмоций. Также проведен наиболее полный обзор акустических признаков. Однако, применение глубокого обучения к распознаванию эмоций в монографии Стейдля рассмотрено не было. На самом деле в нескольких работах были попытки применить глубокие нейронные сети к распознаванию эмоций. Так, например, Kun Han с соавторами в статье [12] предложили двухэтапный подход. Основное отличие от предыдущих работ состоит в том, что нейронная сеть используется не как классификатор, а как генератор признаков. Высказывание, подлежащее классификации, делится на пересекающиеся интервалы, называемые фреймами. Для каждого из них считаются акустические признаки, по которым находятся вероятности быть в одном из эмоциональных состояний. Затем считаются некоторые простые статистики от временного ряда вероятностей, и по ним с помощью специальной нейронной сети принимается решение об эмоции всего высказывания. Одним из недостатков такого подхода, частично решенный в данной работе, является то, что не совсем верно предполагаеть, что эмоциональное состояние фрейма совпадает с состоянием всего высказывания. К сожалению, достаточно сложно учесть влияние положения фрейма в высказывании на его эмоциональное состояние, так как оно будет сильно различаться в зависимости от настоящей эмоции, длительности высказывания, языка и особенностей речи говорящего. В своей работе [13] 2015 года авторы попытались решить описанные выше проблемы. Они оставили двухэтапный подход: сначала использовали рекуррентную нейронную сеть для генерации высокоуровневых признаков, этим учтя временную динамику, а при оценке эмоции всего высказывания использовали наиболее громкие сегменты. Также на каждой эпохе обучения нейронной сети корректировалась обучающая выборка фреймов с помощью скрытой марковской модели - это позволило частично учесть тот факт, что эмоция всего высказывания не совпадает с эмоциональностью ее конкретного короткого участка.

В большинстве других исследований использовался одноэтапный подход. Он состоит в подсчете акустических признаков для всего высказывания, а затем обучения по ним какой-либо классификационной модели. Например, Stuhlsatz et al. [3] и Kim et al. [5] использовали для классификации глубокую нейросеть, а в работах Eyben et al. [6] и Mower et al. [7] обучалась Support Vector Machine (SVM). Работа Rozgic [8] интересна тем, что в ней помимо акустических признаков использовались также лексиграфические, и было показано, что точность классификации в этом случае увеличивается.

В последние несколько лет набирает популярность другая постановка задачи. Считается, что любую речь можно описать с помощью двух или трех размерных параметров. В англоязычной литературе часто используется VAD-модель [13], в которой используются метрики valence (позитивная-негативная), arousal (неожиданность) и dominance (контролируемость), которые меняются на протяжении всего высказывания. Считается, что в трехмерном VAD-пространстве эмоции образуют области, соответствующие конкретным эмоциям. Главное преимущество предсказания размерных метрик эмоциональной речи состоит в большей информативности по сравнению с дискретными классами. А очевидным недостатком – эти метрики являются субъективными, и их оценка зависит от многих факторов, в том числе случайных. Из-за этого, например, намного сложнее получить размеченную базу.

На самом деле в акустических признаках содержится не вся информация о высказывании. Это иллюстрирует психологический эксперимент Мак-Гурка [9] проведенный в 1976 году. В нем показано, что человек при восприятии речи использует не только звуковые данные, но и визуальные. Это позволяет утверждать, что одних только акустических признаков может быть недостаточно для полноценного машинного анализа речи, в том числе распознавания эмоций. В некоторых работах используется, в числе прочих, анализ видеозаписей лиц людей, произносящих некоторые высказывания, однако, в данной работе используются признаки, полученные исключительно из звукового ряда.

Несмотря на то, что исследованию признаков было посвящено множество работ, вопрос о наиболее эффективном наборе для анализа речи остается открытым. Наряду с выбором модели, он играет немаловажную роль в эффективности методов. В настоящее время известно несколько различных типов числовых характеристик звука:

1. признаки, основанные на длительности, и спектре сигнала;

2. мел-кепстральные признаки;

3. основанные на хромаграмме сигнала.

Мел-кепстром называется преобразование Фурье от специальным образом взвешенного логарифма квадрата спектральной плотности сигнала. Его использование обусловлено особенностями восприятия речи человеком. Имеется ввиду, что субъективное восприятие высоты и громкости звука на самом деле нелинейно по частоте и плотности, мел-кепстр преобразованного сигнала эту нелинейность нивелирует.

Хромаграммой называется эволюция распределения спектральной плотности звука по частоте. Для подсчета признаков обычно используются спектральные плотности внутри 12 частотных полос, соответствующих двенадцати октавам [1].

Временные и спектральные признаки содержат автокорреляцию, энтропию и энергию сигнала, энтропию, среднее и медианное значения спектра, а также его смещение и поток [11].

**1.5 Распознавание настроения по сердцебиению и погоде**

Группа исследователей под руководством Питера Глура из Массачусетского технологического института создала новую систему, которая позволяет людям оценивать свое эмоциональное состояние на протяжении длительных промежутков времени, просто нося умные часы на руке. Они написали приложение для часов Pebble 2 и смартфонов под управлением Android. Приложение на часах собирает данные об уровне активности и сердцебиения пользователя, а на смартфоне данные о его перемещениях, погоде, уровне освещенности вокруг и времени.

Для оценки настроения на основе этих данных авторы использовали алгоритм машинного обучения типа «случайный лес». Исследователи решили представить варианты эмоционального состояния в виде двумерного пространства «активность-удовольствие». Каждая из этих двух величин была разбита на три уровня, и, таким образом, они формировали девять комбинаций, описывающих основные типы эмоций (рисунок 1.1).



Рисунок 1.1 – Комбинаций, описывающих основные типы эмоций

Приложение объединяет множество данных и формирует на их основе эмоцию, которую в данный момент испытывает человек. Четыре раза в день оно сообщает ему эту оценку, после чего пользователь может подтвердить ее или изменить, если она не соответствует его настроению. Разработчики протестировали приложение на 60 добровольцах, которые участвовали в исследовании в течение двух месяцев. За счет применения машинного обучения модель, натренированная на отдельных людях достигла точности определения настроения в 86,68 процента, а общая модель 85,1 процента.

Исследователи из MIT и ранее разрабатывали системы для измерения человеческих эмоций, в том числе и на основе носимых устройств. Недавно они научили умные часы определять эмоциональный тон беседы за счет измерения биометрических показателей владельца, а также записи речи собеседников. А в прошлом году они создали систему, которая за счет радиоволн измеряет пульс и дыхание человека, и на основе этих данных предсказывает его эмоциональное состояние.

**2 ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ**

**2.1 Нейронные сети**

Прототипом для создания нейронных сетей послужили биологические нейронные сети. Две трети всей сенсорной информации, которая к нам попадает, приходит с зрительных органов восприятия. Более одной трети поверхности нашего мозга заняты двумя самыми главными зрительными зонами – дорсальный зрительный путь и вентральный зрительный путь [14].

Дорсальный зрительный путь начинается в первичной зрительной зоне, в темечке и продолжается наверх, в то время как вентральный путь начинается на затылке и заканчивается примерно за ушами (рисунок 2.1). Все важное распознавание образов проходит именно там же, за ушами.

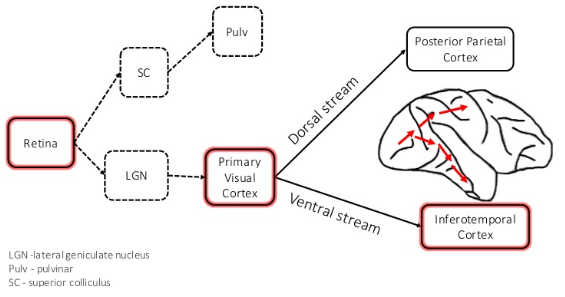


Рисунок 2.1 – Зрительные зоны

Все области, которые используются в нейронных сетях для распознавания образов, пришли именно из вентрального зрительного пути, где каждая маленькая зона отвечает за свою строго определенную функцию.

Изображение попадает к нам из сетчатки глаза, проходит череду зрительных зон и заканчивается в височной зоне.

В 60-е годы прошлого века, когда только начиналось изучение зрительных зон мозга, первые эксперименты проводились на животных, потому что не было фМРТ. Исследовали мозг с помощью электродов, вживлённых в различные зрительные зоны.

Первая зрительная зона была исследована Дэвидом Хьюбелем и Торстеном Визелем в 1962 году. Они проводили эксперименты на кошках. Кошкам показывались различные движущиеся объекты. На что реагировали клетки мозга, то и было тем стимулом, которое распознавало животное. Даже сейчас многие эксперименты проводятся этими драконовскими способами. Но тем не менее это самый эффективный способ узнать, что делает каждая мельчайшая клеточка в нашем мозгу [15].

Таким же способом были открыты еще многие важные свойства зрительных зон, которые мы используем в deep learning сейчас. Одно из важнейших свойств — это увеличение рецептивных полей наших клеток по мере продвижения от первичных зрительных зон к височным долям, то есть более поздним зрительным зонам. Рецептивное поле — это та часть изображения, которую обрабатывает каждая клеточка нашего мозга. У каждой клетки своё рецептивное поле. Это же свойство сохраняется и в нейронных сетях, как вы, наверное, все знаете.

Также с возрастанием рецептивных полей увеличиваются сложные стимулы, которые обычно распознают нейронные сети [16].

На рисунке 2.2 приведены примеры сложности стимулов, различных двухмерных форм, которые распознаются в зонах V2, V4 и различных частях височных полей.

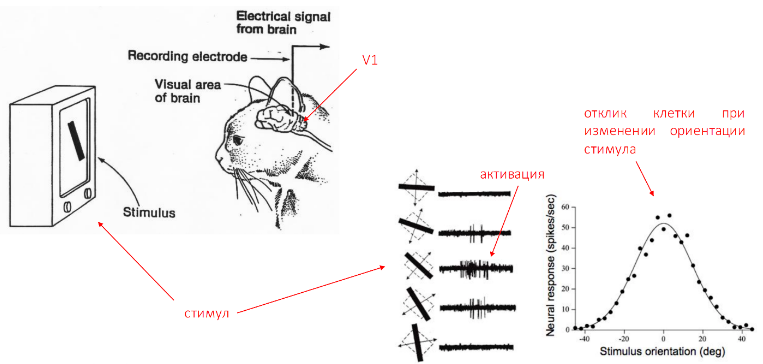


Рисунок 2.2 – Нелинейная модель нейрона

**2.2 Математическая модель нейрона**

Нейрон представляет собой единицу обработки информации в нейронной сети. На рисунке 2.3 показана модель нейрона, лежащего в основе искусственных нейронных сетей. В этой модели можно выделить три основных элемента.

1. Набор синапсов или связей, каждый из которых характеризуется своим весом. В частности, сигнал xi на входе синапса j, связанного с нейроном k, умножается на вес .

2. Сумматор складывает входные сигналы, взвешенные относительно соответствующих синапсов нейрона.

3. Функция активации ограничивает амплитуду выходного сигнала нейрона [17].



Рисунок 2.3 – Нелинейная модель нейрона

Пороговый элемент, который обозначен символом на рисунке 1.1, отражает увеличение или уменьшение входного сигнала, подаваемого на функцию активации.

В математическом представлении функционирование нейрона k можно описать следующей парой уравнений

,

,

где x1, x2, ... , xn – входные сигналы; wk1, wk2, ... , wkm – синоптические веса нейрона k; – линейная комбинация входных сигналов; – порог; – функция активации; – выходной сигнал.

**2.3 Функция активации**

Функция активации представленная как , определяет выходной сигнал нейрона в зависимости от значения сумматора . На рисунке 2.4 изображена сигмоидальная функция активации.

Примером сигмоидальной функции может служить логистическая функция, задаваемая следующим выражением:

,

где – параметр наклона сигмоидальной функции. В пределе, когда параметр наклона достигает бесконечности, сигмоидальная функция вырождается в пороговую [17].



Рисунок 2.4 – Сигмоидальная функция для различных параметров

**2.4 Многослойный персептрон**

Многослойные нейронные сети прямого распространения (рисунок 2.5) характеризуются наличием одного или нескольких скрытых слоев, узлы которых называются скрытыми нейронами, или скрытыми элементами. Функция последних заключается в посредничестве между внешним входным сигналом и выходом нейронной сети. Добавляя один или несколько скрытых слоев, мы можем выделить статистики высокого порядка [17].



Рисунок 2.5 – Структурный граф многослойного персептрона

Многослойный персептрон имеет три отличительных признака:

1. каждый нейрон сети имеет нелинейную функцию активации;

2. сеть содержит один или несколько слоев скрытых нейронов, не являющихся частью входа или выхода сети;

3. сеть обладает высокой степенью связанности, реализуемых по средствам синоптических соединений.

Для этого типа сети выделяют два типа сигналов (рисунок 2.6).



Рисунок 2.6 – Направление двух основных потоков сигнала для многослойного персептрона

1. Функциональный сигнал. Это входной сигнал, поступающий в сеть и передаваемый вперед от нейрона к нейрону по всей сети. Такой сигнал достигает конца сети в виде выходного сигнала.

2. Сигнал ошибки. Сигнал ошибки берет свое начало на выходе сети и распространяется в обратном направлении от слоя к слою [17].

**2.5 Процесс обучения**

Определение процесса обучения предполагает следующую последовательность событий:

1. в нейронную сеть поступают сигналы из внешней среды;

2. в результате этого изменяются свободные параметры нейронной сети;

3. после изменения внутренней структуры нейронная сеть отвечает на возбуждение уже иным образом.

Выделяют две основных парадигмы обучения нейронных сетей.

– Обучение с учителем, то есть процесс обучения, при котором обучение происходит путем предоставления сети последовательности обучающих примеров с правильными откликами.

– Обучение без учителя. Парадигма обучения без учителя самим названием подчеркивает отсутствие руководителя, контролирующего процесс настройки ве-совых коэффициентов. При использовании такого подхода не существует маркированных примеров, по которым проводится обучение сети.

Концептуально участие учителя можно рассматривать как наличие знаний об окружающей среде, представленных в виде пар вход-выход. При этом сама среда неизвестна обучаемой нейронной сети.

На основе встроенных знаний учитель может сформировать и передать обучаемой нейронной сети желаемый отклик, соответствующий данному входному вектору. Этот желаемый результат представляет собой оптимальные действия, которые должна выполнить нейронная сеть. Параметры сети корректируются с учетом обучающего вектора и сигнала ошибки. Сигнал ошибки – это разность между желаемым сигналом и текущим откликом нейронной сети. Корректировка параметров выполняется пошагово с целью имитации нейронной сетью поведения учителя. После окончания обучения учителя можно отключить и позволить нейронной сети работать со средой самостоятельно.

Для обучения без учителя можно воспользоваться правилом конкурентного обучения. Например, можно использовать нейронную сеть, состоящую из двух слоев – входного и выходного. Входной слой получает доступные данные. Выходной слой состоит из нейронов, конкурирующих друг с другом за право отклика на признаки, содержащиеся во входных данных.

В простейшем случае нейронная сеть действует по принципу "победитель получает все". При такой стратегии нейрон с наибольшим суммарным входным сигналом "побеждает" в соревновании и переходит в активное соcтояние. При этом все остальные нейроны отключаются [17].

**2.6 Компьютерное зрение**

В случае компьютерного зрения необходимо большее количество скрытых слоёв. Только тогда система будет осмысленно распознавать то, что она видит.

Изображение разбивается либо на пиксели, либо на некие патчи: 2х2, 3х3, 5х5, 11х11 пикселей – как удобно создателям системы, в которой они служат входным слоем в нейронную сеть.

Сигналы с этих входных слоёв передаются от слоя к слою с помощью синапсов, каждый из слоёв имеет свои определенные коэффициенты. Итак, сигналы передаются от слоя к слою пока не будет опознано лицо.

Условно все эти части можно разделить на три класса, обозначим их как X, W и Y, где Х – это наше входное изображение, Y – это набор лейблов, и нам нужно получить весовые коэффициенты.

При наличии Х и Y это, кажется, просто. Однако это очень сложная нелинейная операция, которая не имеет обратной. Даже имея два заданных компоненты уравнения, очень сложно ее вычислить. Поэтому нужно постепенно, методом проб и ошибок, подбором веса W сделать так, чтобы ошибка максимально уменьшилась (рисунок 2.7), желательно, чтобы стала равной нулю [18].

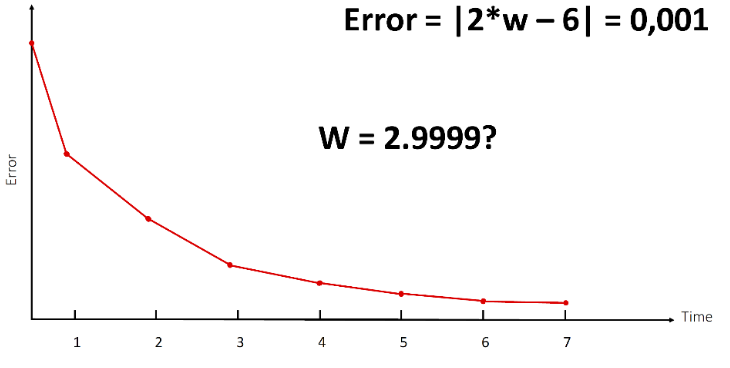
****

Рисунок 2.7 – Оптимизация функции ошибки

Этот процесс происходит итеративно, мы постоянно уменьшаем ошибку, пока не находим то значение веса W, которое нас достаточно устроит.

**2.7 Свёрточная нейронная сеть**

Входное изображение попадает в сеть слоёв, которые можно назвать фильтрами разного размера и разной сложности элементов, которые они распознают (рисунок 2.8). Эти фильтры составляют некий свой индекс или набор признаков, который потом попадает в классификатор. Обычно это либо SVM, либо MLP — многослойный перцептрон.

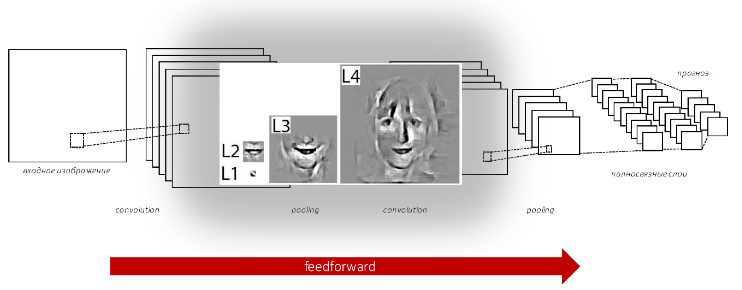


Рисунок 2.8 – Процесс свертки

В свёрточной нейронной сети в операции свёртки используется ограниченная матрица весов небольшого размера, которую «двигают» по всему обрабатываемому слою (в самом начале – непосредственно по входному изображению), формируя после каждого сдвига сигнал активации для нейрона следующего слоя с аналогичной позицией. То есть для различных нейронов выходного слоя используются одна и та же матрица весов, которую также называют ядром свёртки. Её интерпретируют как графическое кодирование какого-либо признака, например, наличие наклонной линии под определённым углом. Тогда следующий слой, получившийся в результате операции свёртки такой матрицей весов, показывает наличие данного признака в обрабатываемом слое и её координаты, формируя так называемую карту признаков. В свёрточной нейронной сети набор весов не один, а целая гамма, кодирующая элементы изображения. При этом такие ядра свёртки не закладываются заранее, а формируются самостоятельно путём обучения сети классическим [методом обратного распространения ошибки](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BE%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D1%80%D0%B0%D1%81%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BE%D1%88%D0%B8%D0%B1%D0%BA%D0%B8" \o "Метод обратного распространения ошибки). Проход каждым набором весов формирует свой собственный экземпляр карты признаков, делая нейронную сеть многоканальной[18]

Если рассмотреть на примере распознавания лиц, то рецептивное поле первого слоя будет маленьким, потом чуть побольше и так до тех пор, пока наконец мы не сможем распознавать уже лицо целиком (рисунок 2.9).

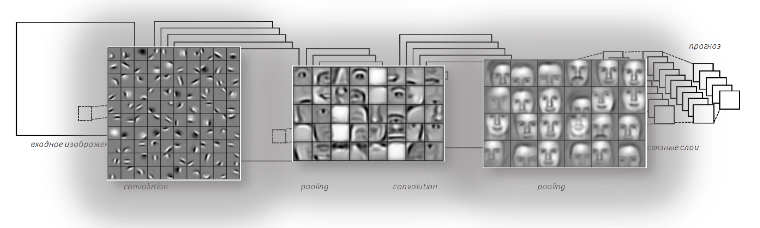
****

Рисунок 2.9 – Рецептивные поля слоев сети

С точки зрения того, что находится у нас внутри фильтров, сначала будут наклонные палочки плюс немного цвета, затем части лиц, а потом уже целиком лица будут распознаваться каждой клеточкой слоя.

**2.8 Рекуррентные нейронные сети**

Рекуррентная нейронная сеть отличается от сети прямого распространения наличием, по крайней мере, одной обратной связи. Наличие обратных связей в сети, показанной на рисунке 2.10, позволяет использовать информацию о предыдущих событиях для анализа последующих. Благодаря этому появляется возможность обрабатывать серии событий во времени или последовательные пространственные цепочки. Поэтому рекуррентные нейронные сети применимы в таких задачах, где нечто целостное разбито на сегменты, например: [распознавание речи](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%80%D0%B5%D1%87%D0%B8" \o "Распознавание речи) или эмоций.

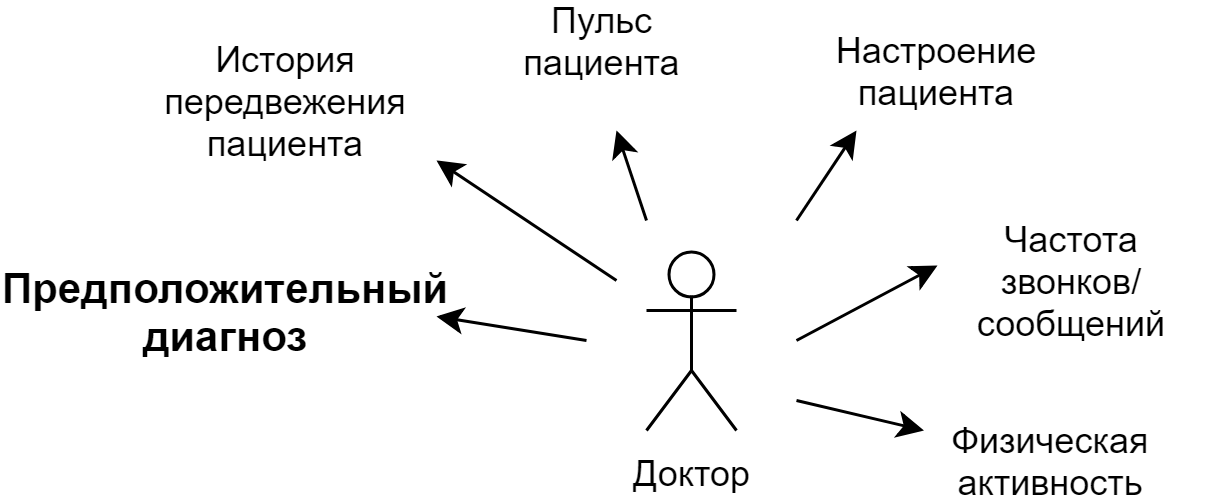


Рисунок 2.10 – Рекуррентная сеть со скрытыми нейронами

Даже в улыбке – одной из самых простых эмоций – есть несколько моментов: от нейтрального выражения лица до того момента, когда будет полная улыбка. Они идут друг за другом последовательно. Чтоб это хорошо понимать, нужно уметь наблюдать за тем, как это происходит, передавать то, что было на предыдущем кадре в следующий шаг работы системы.

**2.8 Сбор поведенческой информации**





**3 ЭКСПЕРИМЕНТАЛЬНЫЙ РАЗДЕЛ**

**3.1 Проектирование системы**

В этой главе будет спроектирована архитектура системы, а также база данных для приложения. Также, необходимо рассмотреть используемые технологии, которые будут применяться при разработке данного программного обеспечения. Существует огромное количество языков программирования и технологий, выбор наиболее удовлетворительных при заданных условиях не всегда оказывается простым. Необходимо рассмотреть все преимущества и недостатки, и, применительно к данной задаче, использовать наиболее подходящие технологические решения.

Клиент-сервер (англ. Client-server) – вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Физически клиент и сервер – это программное обеспечение. Обычно они взаимодействуют через компьютерную сеть посредством сетевых протоколов и находятся на разных вычислительных машинах, но могут выполняться также и на одной машине. Программы расположенные на сервере ожидают от клиентских программ запросы и предоставляют им свои ресурсы в виде данных (например, загрузка файлов посредством HTTP, FTP, BitTorrent или потоковое мультимедиа) или сервисных функций (например, работа с электронной почтой, общение посредством систем мгновенного обмена сообщениями, просмотр веб-страниц во всемирной паутине) [5].

Преимущества:

* Отсутствие дублирования кода программы-сервера программами-клиентами.
* Так как все вычисления выполняются на сервере, то требования к компьютерам, на которых установлен клиент, снижаются.
* Все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов. На сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.

Недостатки:

* Неработоспособность сервера может сделать неработоспособной всю вычислительную сеть. Неработоспособным сервером следует считать сервер, производительности которого не хватает на обслуживание всех клиентов, а также сервер, находящийся на ремонте, профилактике и т.п.
* Поддержка работы данной системы требует отдельного системного администратора.
* Высокая стоимость оборудования.

Трехуровневая архитектура, или трехзвенная архитектура (англ. three-tier или англ. Multitier architecture) – архитектурная модель программного комплекса, предполагающая наличие в нём трёх компонентов: клиентского приложения (обычно называемого «тонким клиентом» или терминалом), сервера приложений, к которому подключено клиентское приложение, и сервера базы данных, с которым работает сервер приложений.

Клиент – это интерфейсный (обычно графический) компонент, который представляет первый уровень, собственно приложение для конечного пользователя. Первый уровень не должен иметь прямых связей с базой данных (по требованиям безопасности), быть нагруженным основной бизнес-логикой (по требованиям масштабируемости) и хранить состояние приложения (по требованиям надежности). На первый уровень может быть вынесена и обычно выносится простейшая бизнес-логика: интерфейс авторизации, алгоритмы шифрования, проверка вводимых значений на допустимость и соответствие формату, несложные операции (сортировка, группировка, подсчет значений) с данными, уже загруженными на терминал.

Сервер приложений располагается на втором уровне. На втором уровне сосредоточена большая часть бизнес-логики. Вне его остаются фрагменты, экспортируемые на терминалы (см. выше), а также погруженные в третий уровень хранимые процедуры и триггеры [5].

Сервер базы данных обеспечивает хранение данных и выносится на третий уровень. Обычно это стандартная реляционная или объектно-ориентированная СУБД. Если третий уровень представляет собой базу данных вместе с хранимыми процедурами, триггерами и схемой, описывающей приложение в терминах реляционной модели, то второй уровень строится как программный интерфейс, связывающий клиентские компоненты с прикладной логикой базы данных.

Достоинства трёхзвенной архитектуры клиент-сервер:

* масштабируемость;
* конфигурируемость – изолированность уровней друг от друга позволяет (при правильном развертывании архитектуры) быстро и простыми средствами переконфигурировать систему при возникновении сбоев или при плановом обслуживании на одном из уровней;
* высокая безопасность;
* высокая надёжность;
* низкие требования к скорости канала (сети) между терминалами и сервером приложений;
* низкие требования к производительности и техническим характеристикам терминалов, как следствие снижение их стоимости.

Терминалом может выступать не только компьютер, но и, например, мобильный телефон.

Недостатки:

* более высокая сложность создания приложений;
* сложнее в разворачивании и администрировании;
* высокие требования к производительности серверов приложений и сервера базы данных, а, значит, и высокая стоимость серверного оборудования;
* высокие требования к скорости канала (сети) между сервером базы данных и серверами приложений.

**3.2 Архитектура системы**

Программное средство состоит из 5 элементов (см. рисунок 3.2):



Рисунок 3.2 – Архитектура системы

1. Мобильное приложение, которое устанавливается на телефон пациента, для сбора поведенческой информации. Было написано с использованием языка Java для мобильной платформы Android. Приложение аккомулирует и отправляет всю информацию о пациенте на веб-сервис.

2. Веб-сервис для получения и предоставления данных о пациенте. Был разработан с использованием технологии ASP.NET Web API 2.0, доступ к ресурсам которого предоставляется через HTTP запросы. Получает всю поведенческую информацию пациента от мобильного приложения и сохраняет её в базу данных.

3. Вычислительный сервис для обработки больших данных пациента. Был разработан с использованием технологии .NET. При поступлении новых данных о пациенте производит сложные вычисления для предсказаний возможного психологического состояния. Запускается на отдельной машине, чтобы снять нагрузку с основного сервиса.

4. Сервер базы данных является хранилищем всей информации о пациенте и пользователях системы.

5. Браузерное приложение, которое позволяет врачу получить доступ к поведенческой информации конкретного пациента, а также увидеть предсказанное системой его психологическое состояние. Приложение было написано с использованием технологий ReactJS, HTML 5.0 и CSS 3.0.

## **3.3** **Проектирование базы данных организационного модуля**

На рисунке 3.3 можно увидеть схему базы данных, спроектированную для хранения всей необходимой информации в системе.

****

Рисунок 3.3 – Схема базы данных

User – сущность пользователя системы. Хранит в себе логин пользователя, хеш пароля, соль пароля, адрес электронной почты.

Role – хранит имя пользовательской роли (Patient/Doctor);

UserRole – промежуточная сущность связи «многие-ко-многим». Хранит роли конкретного пользователя в системе;

Contact – сущность контактной информации ползователя. Хранит в себе страну, язык, адрес, город, почтовый индекс, имя и фамилию пользователя.

Diagosis – хранит имя пользовательской роли (Patient/Doctor);

UserDiagosis – промежуточная сущность связи «многие-ко-многим». Хранит историю диагнозов пациента.

UserCall – сущность звонка пользователя. Хранит в себе ссылку на пользователя, дату и длительность звонка, имя кому был адресован звонок, имя от кого исходил звонок.

UserMessage – сущность сообщения пользователя. Хранит в себе ссылку на пользователя, дату сообщения, имя адресата и адресанта сообщения.

UserLocation – сущность геолокации пользователя. Хранит в себе ссылку на пользователя, дату сообщения, имя адресата и адресанта сообщения.

**3.4 Мобильное приложение**

Основной целью мобильного приложения является сбор поведенческой информации пациента, такой как история звонков, сообщений и перемещений. Данная часть системы не нуждается в наличие графического интерфейса и должна работать в фоне. Поэтому было принято решение создать Android службу (сервис).

Службы (Сервисы) в Android работают как фоновые процессы и представлены классом android.app.Service. Они не имеют пользовательского интерфейса и нужны в тех случаях, когда не требуется вмешательства пользователя. Сервисы работают в фоновом режиме, выполняя сетевые запросы к веб-серверу, обрабатывая информацию, запуская уведомления и т.д. Служба может быть запущена и будет продолжать работать до тех пор, пока кто-нибудь не остановит её или пока она не остановит себя сама. Сервисы предназначены для длительного существования, в отличие от активностей. Они могут работать, постоянно перезапускаясь, выполняя постоянные задачи или выполняя задачи, требующие много времени.

Android даёт службам более высокий приоритет, чем бездействующим активностям, поэтому вероятность того, что они будут завершены из-за нехватки ресурсов, заметно уменьшается. Если система должна преждевременно завершить работу запущенного сервиса, он может быть настроен таким образом, чтобы запускаться повторно, как только станет доступно достаточное количество ресурсов. В крайних случаях приоритет сервиса может быть повышен до уровня активности, работающей на переднем плане.

Запущенные сервисы всегда имеют больший приоритет, чем бездействующие или невидимые активности, поэтому менее вероятно, что их работа завершится преждевременно при распределении ресурсов. Единственная причина, почему Android может досрочно остановить Сервис, – выделение дополнительных ресурсов для компонентов, работающих на переднем плане (как правило, для активностей). Если такое случится, сервис автоматически перезапустится, когда будет достаточно доступных ресурсов.

Чтобы определить службу, необходимо создать новый класс, расширяющий базовый класс Service:

public class BehavioralService extends Service {

public BehavioralService () {…}

…

}

Также необходимо зарегистрировать сервис в манифесте приложения в секции application:

<service

android:name=".BehavioralService"

android:enabled="true"

android:exported="true" >

</service>

На рисунке 3.4.1 можно увидеть жизненный цикл Android службы:



Рисунок 3.4.1 – жизненный цикл Android службы

Жизнь службы начинается с вызова метода onCreate, внутри которого производим начальную инициализацию. Первое, что необходимо сделать, это получить доступ к идентификатору/имени пользователя учетной записи Google через Android:

public String getUsername() {

AccountManager manager = AccountManager.get(this);

Account account = manager.getAccountByType("com.google");

String email = account.name.get(0);

return email.split("@");

}

Активная жизнь службы начинается с вызова метода onStart. Внутри данного метода производим запуск процесса, который будет периодически отправлять поведенческую информацию пациента на сервер:

Boolean stop = false;

new android.os.Handler().postDelayed(

new Runnable() {

public void run() {

if (!stop)

behaviourData.sync();

}

}, 5000);

При остановке работы службы вызывается метод onDestroy, где выполняется освобождение всех ресурсов. В теле метода устанавливаем флаг stop в значение true. В результате чего будет остановлена синхронизации данных с сервером.

@Override

public void onDestroy() {

stop = true;

}

Для доступа к истории звонков и сообщений был использован поставщик контента Android. Поставщик контента предоставляет данные внешним приложениям в виде одной или нескольких таблиц, аналогичных таблицам в реляционной базе данных. Строка представляет собой экземпляр некоторого типа собираемых поставщиком данных, а каждый столбец в этой строке – это отдельный элемент данных, собранных для экземпляра.

В листинге ниже, используя массив из ключей, указываем какую информацию о звонках нам необходимо получить:

String[] projection = new String[]{

CallLog.Calls.\_ID,

CallLog.Calls.DATE,

CallLog.Calls.NUMBER,

CallLog.Calls.CACHED\_NAME,

CallLog.Calls.DURATION,

};

Для доступа к данным из поставщика контента используется клиентский объект [ContentResolver](https://developer.android.com/reference/android/content/ContentResolver.html?hl=ru):

Cursor cursor = ContentResolver.query(

CallLog.Calls.CONTENT\_URI,

projection,

);

Таким же образом запрашиваем историю сообщений:

String[] projection = new String[]{

SmsLog.Sms.\_ID,

SmsLog.Sms.DATE,

SmsLog.Sms.NUMBER,

SmsLog.Sms.CACHED\_NAME,

};

Cursor cursor = ContentResolver.query(

SmsLog.Sms.CONTENT\_URI,

projection,

);

Также необходимо запросить разрешение на чтение истории звонков и сообщение в манифесте приложения:

<uses-permission android:name="android.permission.READ\_CALLS" />

<uses-permission android:name="android.permission.READ\_SMS" />

Смартфоны и планшеты на базе Android постоянно фиксируют местоположение пользователя и хранят эти данные. Координаты они определяют несколькими способами: через спутники GPS, башни сотовой связи или точки Wi-Fi. История хранится с самого первого дня, когда у пользователя появилось устройство на Android, и не очищается автоматически.

Android предоставляет приложениям доступ к службам определения местоположения с помощью классов в пакете android.location. Центральным компонентом структуры местоположения является системная служба LocationManager, которая предоставляет API-интерфейсы для определения местоположения и ориентации устройства (если доступно).

В листинге ниже мы запрашиваем экземпляр LocationManager из системы, используя который, мы получаем последнее зарегистрированное телефоном местоположение пользователя:

mLastLocation = [getSystemService(Context.LOCATION\_SERVICE)](https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.String))

.getLastLocation(mGoogleApiClient);

store.saveLocation(mLastLocation)

Google Maps не предоставляет историю перемещений пользователя, так как это нарушает правила конфиденциальности. Поэтому нам доступно только последнее зарегистрированное телефоном местоположение.

Когда пользователь первый раз открывает приложение, он видит экран авторизации (см. рисунок 3.4.2). Пользователю предлагается пройти авторизацию с использованием Google аккаунта.

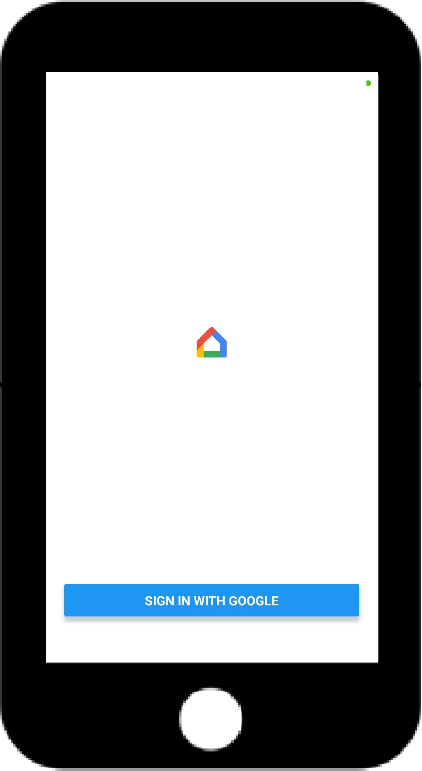


Рисунок 3.4.2 – Экран авторизации

При нажатии на кнопку «SIGN IN WITH GOOGLE», приложение отправляет запрос на google-сервис, чтобы получить токен для доступа к информации пользователя. Далее происходит перенаправление пользователя на страницу авторизации google. Если пользователь имеет несколько аккаунтов, google предлагает выбрать один из имеющихся (см. рисунок 3.4.3).

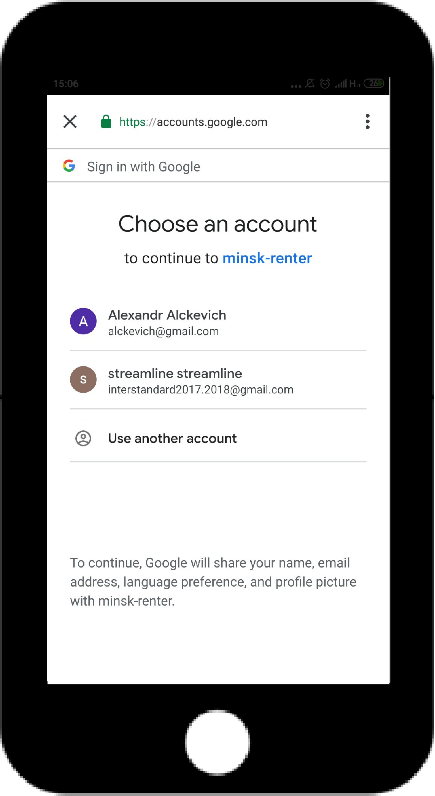


Рисунок 3.4.3 – Страница авторизации google

Когда пользователь успешно проходит авторизацию Google, наше прилодение получает токен доступа к его информации. Используя токен, мы можем запросить адрес электронного почтового ящика, чтобы зарегистрировать пользователя в системе.

После того как пользователь проходит успешно авторизацию, запускается Android служба, которая работает в фоне.

**3.5 Сервер**

Для реализации сервера была использована технология ASP.Net Web API 2 на платформе .NET Framework 4.5. Для доступа к объектам реляционной базы данных в зависимости проекта был добавлен Entity Framework версии 6 с помощью NuGet package manager.

Для создания API-интерфейсов, которые предоставляют службы, и данные было реализовано несколько объектов-контроллеров, которые обрабатывают HTTP-запросы. На рисунке 3.5.1 можно увидеть контроллеры, которые определяют контракт доступа к данным сервера.



Рисунок 3.5.1 – Серверные контроллеры

Для представления данных в приложении были созданы следующие объекты-модели:

a) CallModel – представляет данные о звонке пациента;

b) SmslModel – представляет данные о сообщении пациента;

c) LocationModel – представляет данные о местоположении пациента;

b) PatientListModel – модель представления пациента, используется для вывода списка пациентов в веб-приложении для доктора.

Объекты представления сериализуются в формат JSON, а затем записываются в тело HTTP запроса/ответа.

BehaviorController перехватывает HTTP запросы со стороны клиента, проверяет пришедшие данные и, если данные валидны, перенаправляет обработку запроса объекту реализующему интерфейс IBehaviorService. На рисунке 3.5.2 можно увидеть серверные сервисы.



Рисунок 3.5.2 – Серверные сервисы

Для сохранения поведенческой информации пациента необходимо по маршруту serverUrl/api/behavior отправить HTTP запрос с POST методом, в теле которого должна быть размещена вся необходимая информация в формате JSON.

PatientListController наследуется от ODataController, что позваляет, при обращении к API контроллера использовать возможности протокола OData (Open Data Protocol). OData расширяет RESTful API стандарт и позволяет клиентам формировать запрос, используя параметры фильтрации, разбиения на страницы, сортировки данных.

Для реализации аутентификации между клиентским и серверным приложением был использован подход, основанный на токене.

Когда пользователь успешно выполняет вход в систему со своими учетными данными, сервер использует секретный ключ для шифрования необходимых пользовательских данных и генерирует, на основе этих данных, токен. Этот токен возвращается клиенту и должен быть сохранен в локальном хранилище браузера или в cookies.

Всякий раз, когда пользователь захочет получить доступ к защищенному ресурсу, он должен отправить токен в заголовке авторизации с использованием схемы Bearer (Authorization: Bearer <token>). Затем сервер использует секретный ключ для дешифрования и проверки токена, и если он действителен, пользователь получает доступ к защищенным ресурсам.

Аутентификация на основе токенов является механизмом аутентификации без состояния, поскольку пользовательское состояние никогда не сохраняется в памяти сервера. Такой подход уменьшает необходимость многократного запроса к базе данных, поскольку токен содержит всю необходимую информацию. Технически, после того, как токен подписан, он действителен навсегда, если явно не установлен срок действия.

Так как модули должны принимать запросы, поступающие из любого источника, был использован механизм CORS (Cross Origin Resource Sharing), который обеспечивает междоменное управление доступом и безопасную междоменную передачу данных.

Для включения механизма аутентификации и защиты данных каждый модуль должен содержать следующую настройку:

public static void ConfigureOAuth(IAppBuilder app, IContainer container)

{

var oAuthServerOptions = new OAuthAuthorizationServerOptions()

{

AllowInsecureHttp = true,

TokenEndpointPath = new PathString("/api/token"),

AccessTokenExpireTimeSpan = tokenexpiration,

Provider = container

.Resolve<CustomAuthorizationServerProvider>(),

RefreshTokenProvider = container.Resolve<RefreshTokenProvider>()

};

// Token Generation

app.UseOAuthAuthorizationServer(oAuthServerOptions);

}

**3.5 Вычислительный сервис**

Сервис производит сложные вичисления над большими данными и сильно загружает систему на которой работает. Поэтому было принято решение запускать его на отдельной от сервера машине.

Так как новая информация о пациенте поступает в базу данных постоянно, было бы не целесообразно проводить вычисления каждый раз при её изменении. Поэтому было принято решение запускать сервис каждые двадцать четыре часа. При запуске производится проверка данных пациента на наличие новой информации, и выполняется пересчет психологического состояния. Для запуска такого рода рекурентной задачи была использована стороння библиотека Hangfire.

В листинге ниже представлен код, который регистрирует

**3.6** **Веб-приложение**

Веб-приложения позволяет врачу получить доступ к поведенческой информации конкретного пациента, а также увидеть предсказанное системой его психологическое состояние. Приложение было написано с использованием технологий ReactJS, HTML 5.0 и CSS 3.0.

Для удобства перемещения и определения места положения на сайте была создана боковая панель навигации. Как можно увидеть на рисунке 3.6.1, когда пользователь переходит на какую-либо страницу, соответствующая секция в навигационной панели выделяется оранжевым цветом. Для пациентов в боковом меню была выделена секция «Patients».



Рисунок 3.6.1 – Боковая панель навигации

При нажатии на кнопку «Patients» в боковой панели навигации, система направляет пользователя на страницу со списком пациентов (см. рисунок 3.6.2).



Рисунок 3.6.2 – Список пациентов

Спмсок пациентов состоит из следующих колонок:

– Number – уникальный номер пациента. Автоматически присваивается пациенту системой при создании;

– Patient Name – имя пациента;

– Age – возраст пациента;

– Sex – пол пациента;

– Mental state – психологическое состояние пациента. Автоматически вычисляется системой на основе поведенческой информации пациента. Как можно увидеть на рисунке 3.6.2, пациенты с вычисленным диагнозом выделяются в списке желтым цветом.

При наведении курсора мыши на строку с пациентом в таблице, строка подсвечивается оранжевым цветом (см. рисунок 3.6.3).



Рисунок 3.6.3 – Подсветка пациента при наведении курсора мыши

При нажатии на строку пациента в таблице, система направляет пользователя на страницу с детальной информацией о пациенте (см. рисунок 3.6.4).



Рисунок 3.6.4 – Cтраница с детальной информацией о пациенте

Страница пациента состоит из следующих частей:

1) Боковая панель – содержит фотографию, имя, возраст пациента и его диагноз, который был автоматически присвоен пациенту системой. Панель остается видна, когда пользователь переключается между вкладками страницы пациента.

2) Вкладка «Detail» – содержит контактную информацию (страна, язык, адрес, город, адрес электронной почты, почтовый индекс) и диагноз пациента.

3) Вкладка «Calls» (см. рисунок 3.6.5) – отображает историю звонков пациента в виде обычного списка и столбчатого графика. При построении столбчатого графика по оси ординат было отложено количество звонков, по оси абсцисс – дни; каждому количеству звонков соответствует столбик. У пользователя есть возможность указать интересуемый временной интервал c помощью полей «From» и «To».



Рисунок 3.6.5 – История звонков пациента

4) Вкладка «Messages» (см. рисунок 3.6.6) – отображает историю сообщений пациента в виде обычного списка и столбчатого графика. При построении столбчатого графика по оси ординат было отложено количество сообщений, по оси абсцисс – дни; каждому количеству сообщений соответствует столбик. У пользователя есть возможность указать интересуемый временной интервал c помощью полей «From» и «To».



Рисунок 3.6.6 – История сообщений пациента

5) Вкладка «GPS» (см. рисунок 3.6.7) – отображает историю перемещений пациента в виде обычного списка и интерактивной карты с метками. У пользователя есть возможность указать интересуемый временной интервал c помощью полей «From» и «To».



Рисунок 3.6.7 – История перемещений пациента

**3.7** **Тестирование системы**

При тестировании системы применялся как ручной, так и автоматизированный подходы. Из автоматизированных видов в системе используется модульное и интеграционное тестирование.

Основные аспекты бизнес-логики приложения покрыты модульными тестами. Как правило, это различные вычислительные операции, обработка данных, управление правами доступами.

Работа с базами данных, а также эмуляция HTTP запросов к веб-приложению покрыты интеграционными тестами. Интеграционные тесты позволяют протестировать создание, обновление, удаление данных. С точки зрения веб-приложения, интеграционные тесты позволяют проверить работоспособность различных обработчиков запросов, правильность формата полученных данных из запроса, правильность возвращаемого сервером ответа.

Пример тестирования указанного пользователем времени прибытия заказа:

[Test]

public void ActualArrivalCanNotBeGreaterThanToday ()

{

jobModel.CurrentStatusId = HallmarkingJobStatus.AwaitingItems;

jobModel.ActualArrival = DateTime.Today.ToUniversalTime()

.AddDays(10);

Assert.ThrowsAsync(async () =>

{

await hallmarkingJobValidationService.ValidateOnSave(jobModel);

});

}

Данный модульный тест проверяет, чтобы указанная пользователем дата прибытия изделий заказа не была меньше текущей даты более чем на один день.

Алгоритм любого теста имеет примерно следующую схему:

1) инициализация сервиса/провайдера – класса, метод которого будет тестироваться;

2) подготовка тестовых данных;

3) передача тестовых данных в тестируемый метод;

4) проверка правильности результата.

Пример интеграционного теста получения данных из таблицы офисов из тестовой базы данных:

[Test]

public async Task ShouldReturnOffices()

{

var context = new CompanyStructureContext();

queryHandler = new GetOfficeListQueryHandler(context);

var query = new GetOfficeListQuery();

var result = await queryHandler.HandleAsync(query);

Assert.That(result.Count(), Is.EqualTo(2));

}

В начале каждого теста удаляется тестовая база данных. Далее создаётся «чистая» база данных, заполняется нужными данными для конкретного тест-кейса.

Интеграционные тесты, как правило, имеют значительно большее время выполнения, чем модульные. Если рассматривать промышленную разработку, то рекомендуется модульные тесты запускать при каждой сборке приложения перед каждой фиксацией изменений кода, а интеграционные – раз в день, либо перед релиз-сборкой на интеграционном сервере.

**ЗАКЛЮЧЕНИЕ**

Реализация интеллектуальной системы, способной следить за психологическим состоянием человека – это решение технически и математически сложной задачи, требующей углубленного изучения предметной области, собственного анализа, умения и навыков при обработке и выборе экспериментальных данных, глубокие знания в области дискретной математики и программировании, а также психологии и многих других областях науки.

Использование нейронных сетей, при разработке систем классификации, позволяет структурировать плохоформализованные данные, ускорить процесс обработки данных, оценить адекватность среды, в которую помещена информационная модель нейронной сети.

Основная цель магистерской диссертации заключалась в создании системы способной в режиме реального времени оценивать психологическое состояние человека, а также в выборе информационной модели нейронной сети и описании алгоритма классификации.

Была выбрана информационная модель на основе многослойного персептрона. Данная модель оптимальна по внутренней структуре и способу управления информационными потоками между нейронами. Кроме того, такая модель способна минимизировать число входных элементов. Модель на основе многослойного персептрона является универсальной моделью и подходит для решения задач разного уровня сложности, в том числе и для решения задачи классификации.

Основные выводы и результаты исследования:

1) На основании изученного материала по искусственным интеллектуальным системам были выявлены наиболее эффективные системы и точные методы классификации, отмечены успешные примеры реализации. Система, которая использовалась в данной работе являлась самообучающейся.

2) Обзор основных видов информационных моделей нейронных сетей позволил выделить две универсальные модели, применимые для широкого круга задач: радиальные нейронные сети прямого распространения и однонаправленные многослойные сети. Выбор информационной модели для решения практической задачи был сделан в пользу однонаправленных многослойных сетей (модель многослойного персептрона). Данная модель подходит по внутренней структуре и математическому описанию и предусматривает разные подходы к решению задачи.

3) В практической части работы была разработана автоматизированная система по распознаванию психологического состояния человека. Система предоставляет возможность в режиме реального времени отслеживать поведение человека через интернет, делать предположения о его психологическом состоянии. Система состоит из мобильного приложения, разработанного под платформу Android, двух веб-сервисов на платформе .NET Framework, одного ReactJs веб-приложения и одной реляционной базы данных.

На основании полученных результатов можно сделать вывод о том, что все поставленные задачи были выполнены в полном объеме, следовательно, главная цель была достигнута.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Манёров В. X., Шнейдер Е. М. Автоматическое распознавание эмоций по спектральным и интонационным признакам // Материалы доклада и сообщения 5-го Всесоюзного совещания-симпозиума цикла «Акустика речи и слуха». Одесса, 1989.

[2] Frijda, N. H. (1986). The emotions. Cambridge: Cambridge University Press.

[3] Big Library Info [Электронный ресурс]. – Режим доступа: http://[www.big-library.info](http://www.big-library.info/).

[4] Определение эмоций по речи [Электронный ресурс]. – Режим доступа: <http://speetech.by/press/analytics/5>.

[5] Определение эмоций по речи [Электронный ресурс]. – Режим доступа: http://www.emotionlabs.ru/content/66/.

[6] Выражение эмоций в языке и речи [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/vyrazhenie-emotsiy-v-yazyke-i-rechi>.

[7] Синтез речи, анализ эмоций и биометрия [Электронный ресурс]. – Режим доступа: https://hh.ru/article/312409.

[8] Распознавание эмоций по речевому сигналу с помощью функций модуляционной теории звуковых сигналов [Электронный ресурс]. – Режим доступа: <http://euroasia-science.ru/tehnicheskie-nauki/raspoznavanie-emocij-po-rechevomu-signalu-s-pomoshhyu-funkcij-modulyacionnoj-teorii-zvukovyx-signalov/>.

[9] Автоматическое распознавание эмоций человека на основе реконструкций аттракторов образцов речи [Электронный ресурс]. – Режим доступа: <https://www.nbpublish.com/library_get_pdf.php?id=21913>.

[10] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” Comput. Netw., vol. 54, no. 15, pp. 2787–2805, 2010.

[11] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, “Convergence of MANET and WSN in IoT urban scenarios,” IEEE Sens. J., vol. 13, no. 10, pp. 3558–3567, Oct. 2013.

[12] A. Laya, V. I. Bratu, and J. Markendahl, “Who is investing in machine-tomachine communications?” in Proc. 24th Eur. Reg. ITS Conf., Florence, Italy, Oct. 2013, pp. 20–23.

[13] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, “Smart cities and the future internet: Towards cooperation frameworks for open innovation,” The Future Internet, Lect. Notes Comput. Sci., vol. 6656, pp. 431–446, 2011.

[14] Уоссермен, Ф. Нейрокомпьютерная техника:  Теория и практика: пер. с англ. Ю. А. Зуев / В. А. Точенов – М. : Феникс, 1992. – 184 с.

[15] Научная библиотека избранных естественно научных изданий [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://sernam.ru/book_kir.php?id=25>.

[16] Цифровая обработка изображений [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.sibsauktf.ru/courses/fulleren/g3.htm>.

[17] Хайкин, Саймон, Нейронные сети: полный курс, 2-е изд., испр. : Пер. с англ. - М. : ООО "И.Д. Вильямс 2006. – 1104с.

[18] Распознавание текста в ABBYY FineReader [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://habrahabr.ru/company/abbyy/blog/225215>.

[19] Методы распознавания текста [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://habrahabr.ru/post/220077>.

[20] Нейронные сети [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.aiportal.ru/articles/neural-networks/1/>.

[21] Бинаризация изображений а [Электронный ресурс]. – Электронные данные. – Режим доступа: http://recog.ru/blog/applied/15.html.

[22] Система кодирования лицевых движений [Электронный ресурс]// основные мимические выражения/ URL: http://dic.academic.ru/dic.nsf/ruwiki/1628394 (дата обращения 16.04.17).

[23].Эмоциональная напряженность методика определения [Электронный ресурс]/URL: http://pandia.ru/text/80/079/8545.php (дата обращения 19.04.2017).

[24] Татаренков Д.А. Анализ методов обнаружения лиц на изображении//Технические науки. Молодой ученый. 2015. №4(84).

[25] Халафян А.А. Статистический анализ данных. 3-е изд. учеб./ Бином – Пресс. 2007. 512 с. 40. Царегородцев В. Г. Вычислительные технологии// Вестник КазНУ. ч 3. 2008. с. 308-315.

[26] Использование Angular.js на вебсайтах [Электронный ресурс]. – Режим доступа: https://angularjs.org/.

[27] Новые элементы в HTML 5 [Электронный ресурс]. – Режим доступа: <http://www.ibm.com/developerworks/ru/library/x-html5>.

[28] Бер Бибо, jQuery. Подробное руководство по продвинутому JavaScript: Символ-Плюс, 2009. – 384c.

[29] Code First в Entity Framework [Электронный ресурс]. ‑ Режим доступа: <http://metanit.com/sharp/entityframework/1.2.php>.

[30] Рихтер Дж., CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. – Пер. с англ./ Дж. Рихтер – СПб: Питер, 2015. – 896 с.