

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование кафедры)

01.03.02 Прикладная математика и информатика

(код и наименование направления подготовки, специальности)

Системное программирование и компьютерные технологии

(направленность (профиль)/специализация)

## БАКАЛАВРСКАЯ РАБОТА

на тему **Дифференциация меланомы среди злокачественных новообразований кожи на спектрах комбинационного рассеяния с применением искусственных нейронных сетей**

Студент	<u>И.В. Стебунов</u> (И.О. Фамилия)	_____ (личная подпись)
Руководитель	<u>С.В. Баумгертнер</u> (И.О. Фамилия)	_____ (личная подпись)
Консультанты	<u>Н.В. Андрюхина</u> (И.О. Фамилия)	_____ (личная подпись)

**Допустить к защите**

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский  
(ученая степень, звание, И.О. Фамилия) \_\_\_\_\_ (личная подпись)  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г.

Тольятти 2019

## АННОТАЦИЯ

Рассматривается проблема неинвазивной диагностики раковых заболеваний тканей кожи человека. Проводится анализ результатов спектроскопии комбинационного рассеяния (КР), для выявления меланомы среди кожных патологий с использованием искусственных нейронных сетей. Основой для исследования стали методики дифференциации патологий, обосновывающие наличие в спектрах КР классифицирующих признаков.

Для решения данной задачи были разработаны несколько нейронных сетей, состоящих из одного и двух скрытых слоев и обученных с помощью алгоритма обратного распространения ошибки. Анализируя спектры, были получены признаки пригодные для обучения многослойного перцептрона.

Входные данные представляется в виде каскада горизонтально расположенных частей спектра, сохраняя зависимость значений интенсивности от длины волны, позволяя нейронной сети самой составлять карты признаков, отвечающие за дифференциацию меланомы.

Эффективным решением в этой области может стать использование высокоуровневого языка программирования Python 3.6, предоставляющий широкий инструментарий для решения задач машинного обучения и использования сторонних библиотек Keras, Scikit-learn. Созданные на их базе архитектуры были обучены на подготовленных наборах данных. Для обобщения результатов обучения был разработан алгоритм принятия решения.

Работа состоит из введения, трех глав и заключения. Объем работы составляет 51 страниц, на которых размещены 25 рисунка, 5 таблицы и 2 приложения. При написании данной работы было использовано 23 литературных источников, в том числе 9 источников на иностранном языке.

## **ABSTRACT**

The title of the bachelor's thesis is «The differentiation of pathologies in Raman spectra using artificial neural networks».

The object of the bachelor's thesis is the methods of pathology differentiation, which substantiate the presence of classifying signs in the Raman spectra. The subject of the thesis is artificial neural networks.

The bachelor's thesis deals with problems of noninvasive diagnostics of oncological diseases of human skin tissues. The analysis of the results of Raman spectroscopy scattering is carried out to identify signs of differentiation of cutaneous pathologies using a neural network.

First we consider the study of the possibility of learning a neural network on the Raman spectra. The basis for the study was the stochastic methods for the differentiation of pathologies, which substantiates the presence in the OP of the spectral features of the pathologies classification. Then we explore the neural networks. With this approach, the neural network itself creates a map of the signs responsible for the differentiation of pathology. Much attention is paid to the processing of input data. We also report on the results of experiments to identify skin pathologies using neural networks.

An effective solution in this area can be the usage of the high-level programming language Python 3.6 and the libraries Keras, Scikits-learn. The architectures created on their basis were trained on the prepared data sets.

Based on the learning outcomes, a weighted decision making algorithm was developed. The final work is on 51 pages, an introduction that includes 25 figures, 5 tables, a list of 23 references, including 9 foreign sources and 2 appendices.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
1 АНАЛИЗ ПРОБЛЕМЫ РАКА КОЖИ И ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ .....	8
1.1 Рак кожи и методы его диагностики .....	8
1.2 Спектроскопия видимого и ультрафиолетового света, Раман-спектроскопия.....	11
1.3 Общая характеристика искусственных нейронных сетей .....	13
2 ПРОЕКТИРОВАНИЕ НЕЙРОСЕТЕВЫХ АРХИТЕКТУР .....	16
2.1 Постановка задачи для дифференциации меланомы кожи .....	16
2.2 Подготовка данных и обучение искусственной нейронной сети .....	17
2.3 Задача обучения искусственной нейронной сети .....	18
2.4 Проблемы обучения искусственной нейронной сети .....	24
2.5 Альтернативные методы обратного распространения ошибки .....	27
2.6 Разработка архитектуры и программная реализация искусственной нейронной сети .....	30
2.7 Перекрестная проверка и подбор параметров по сетке .....	36
2.8 Программа принятия решения и ее блок-схема.....	41
3 ТЕСТИРОВАНИЕ НЕЙРОСЕТЕВЫХ АРХИТЕКТУР .....	42
3.1 Разделение данных и их нормализация .....	42
3.2 Методы оценки точности работы искусственной нейронной сети .....	43
3.3 Тестирование искусственных нейронных сетей и программы принятия решений.....	44
ЗАКЛЮЧЕНИЕ .....	48
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ .....	49
ПРИЛОЖЕНИЕ А .....	52

## ВВЕДЕНИЕ

Кожа человека является очень важной частью организма, выполняя функцию барьера между организмом и внешней средой. Защищая человеческий организм, она постоянно подвергается вредному воздействию внешней среды, что в конечном счете может привести к кожным заболеваниям или раку кожи. Наиболее распространёнными факторами, способствующим возникновению рака кожи, является длительное облучение солнечными лучами ультрафиолетового спектра, радиоактивного излучения, длительное термическое воздействие, контакт с мышьяком, смолами, дегтем, сажей [1].

Рак кожи — это эпителиальные злокачественные новообразования, среди которых собственно рак кожи составляет до 90% [1]. Выделяют несколько форм рака кожи, базально-клеточный рак кожи — наименее злокачественный вид опухоли. плоскоклеточный рак кожи, и меланома — эта форма патологии обладает высокой степенью злокачественности и в большинстве случаев ведет к летальному исходу.

Злокачественная опухоль на ранних стадиях в явной форме тяжела для диагностирования и с течением времени увеличивается в размерах, и чем позже происходит ее обнаружение, тем сложнее будет проходить процесс лечения, и возможно появление рецидивов заболевания и образование метастазов. При обнаружении рака начальной стадии борьба с опухолью наиболее эффективна, вместе с тем очень высока вероятность выздоровления.

Основной задачей для предотвращения летального исхода является необходимость диагностировать заболевание на ранних стадиях, но не всегда лечащий врач, который не специализируется в области дерматологии или онкологии, в состоянии правильно поставить диагноз. Классическая диагностика рака кожи включает гистологический метод и цитологию мазков, позволяющие верифицировать диагноз на основании обнаружения специфических патоморфологических признаков [6]. Но проблема

заключается в том, что все они требуют непосредственно присутствие человека при анализе полученных данных. От чего появляется риск человеческой ошибки при постановке диагноза. Чтобы исключить возможность ошибки возможно применение современных технологий компьютерного анализа.

В данной работе будет рассмотрена одна из таких технологий, основанная на дифференциации меланомы среди патологий на спектрах комбинационного рассеяния с применением нейронных сетей, так как нейронные сети хорошо себя показали в дифференциации патологий [4].

Целью данной работы является повышение точности диагностики меланомы кожи с применением спектров комбинационного рассеяния с помощью разработки архитектуры классификатора с использованием нейронных сетей.

Объектом исследования является процесс ранней компьютерной диагностики кожи на спектрах комбинационного рассеяния.

Предметом исследования является архитектура нейронной сети для эффективного классифицирования заболеваний меланомы кожи на спектрах комбинационного рассеяния.

Актуальность работы заключается в том, что за 2007-2017 гг. численность контингента больных меланомой кожи на 100 000 населения выросла с 41,2 до 61,2, и ранняя диагностика без вмешательства врачей с использованием современных технологий сокращает случаи несвоевременно замеченных кожных заболеваний, в частности меланомы, так как она является самым опасным и прогрессивным кожным заболеванием [5].

При выполнении бакалаврской работы использовался высокоуровневый язык программирования Python 3.6, предоставляющий широкий инструментарий для решения задач машинного обучения. Обучение проводилось по данным результатов спектроскопии с помощью нейронных сетей.

Для достижения поставленной цели, в работе решался ряд задач:

1. Изучение существующих литературных источников, описывающих передовые достижения в области разработки высокоточных моделей классификации, основанных на нейронных сетях. Обобщение информации, полученной о нейронных сетях, раке кожи, спектроскопии.

2. Построения наиболее эффективной архитектуры нейронной сети и проектирование адаптивной высокоточной модели классификатора с использованием нейронных сетей.

3. Обучение полученной нейронной сети.

4. Проведение экспериментов по сравнению качества результатов классификации предложенных моделей, оценка и интерпретация результатов.

5. Расчет, визуализация и интерпретация сравнительных данных о результатах проведенных экспериментов.

Выпускная квалификационная работа состоит из введения, трех глав и заключения.

В первой главе обосновывается актуальность проделанной работы, представляются теоретические сведения о раке кожи, диагностики спектров комбинационного рассеяния и методы обучения искусственных нейронных сетей.

Во второй главе описываются нейросетевые модели для классификации патологий, разрабатывается и программируется архитектура сети, также подготавливаются наборы данных для их обучения, производится подбор параметров и обучение сети.

В третьей главе производится тестирование обученных нейросетевых архитектур на основе подготовленных наборов данных.

В заключении подводятся итоги исследования, формируются окончательные выводы по рассматриваемой теме.

# 1 АНАЛИЗ ПРОБЛЕМЫ РАКА КОЖИ И ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

## 1.1 Рак кожи и методы его диагностики

Рак кожи — это группа опухолей с локализацией на коже. Рак кожи занимает одно из лидирующих мест в структуре онкологических заболеваний мужского и женского населения в Республике Беларусь, Российской Федерации, а также в странах Западной Европы и обусловлен развитием аномальных клеток, которые способны проникать или распространяться на другие части тела [7].

Основной причиной, способствующей возникновению рака кожи, является длительное нахождение на солнце и облучение ультрафиолетовыми лучами. Еще одним фактором может стать радиоактивное излучение, термическое воздействие, контакт с мышьяком, смолами, дегтем, сажей [7].

В 2017 г. в России доля снятых с учета больных базально-клеточным раком кожи от впервые взятых больных раком кожи (кроме меланомы) составила 53,4% (2016 г. – 62,1%; 2015 г. – 64,1%, 2014 г. – 61,7%). Разброс данного показателя по территориям России колеблется от 9,0% до 198,4% [5].

Среди различных типов злокачественных новообразований наиболее распространенными являются меланома, базальноклеточный рак, плоскоклеточный рак:

- Базально-клеточный рак кожи является наименее опасным злокачественным новообразованием на коже. Первичный морфологический элемент базалиомы – узелок, обладающий клиническими характеристиками, такими как небольшой размер около 1-5 миллиметров полушаровидной формой, восковидным блеском гладкой поверхности, покрытой телеангиэктазиями, застойно-розовой или розовато-бурой окраской с перламутровым оттенком, плотной консистенцией, отсутствием субъективных ощущений. Выделяют узловую (узелковую), поверхностную, склеродермоподобную, язвенную, рубцующуюся и пигментную формы [2].

- Плоскоклеточный рак кожи: встречается в 25% и менее случаях, часто является причиной ультрафиолетовых ожогов на открытых участках кожи, например, солнечными лучами. Плоскоклеточный рак — злокачественная опухоль, развивающаяся из кератиноцитов и способная вырабатывать кератин — встречается реже, часто возникает на фоне предраковых заболеваний кожи. Опухоль обычно одиночная, может располагаться на любых участках тела. От базалиомы отличается быстрым инфильтрирующим ростом и способностью метастазировать. Метастазирует главным образом лимфогенным путем [9]. У больных плоскоклеточным раком кожи размером менее двух сантиметров пятилетняя выживаемость составляет около 90%, при больших размерах и прорастании опухоли в подлежащие ткани — менее 50% [8].

- Меланома: регистрируется в 2% случаях и является патологией с высокой степенью злокачественности. Меланома является злокачественной опухолью нейроэктодермального происхождения, развивающейся из меланоцитов, расположенных преимущественно в коже. Поэтому кожная форма заболевания, а именно меланома кожи, является самой частой, в то время как на долю внекожных форм заболевания, возникающих на слизистых прямой кишки и гениталий, полости рта и носовых пазух, а также оболочек головного и спинного мозга приходится всего лишь 1-3%. Еще около 7% приходится на глазную форму заболевания, при которой чаще всего поражается сетчатка и значительно реже - конъюнктура глаза [2]. Характерен быстрый переход от 1 до 4 стадии. Шанс выживаемости стремительно падает с каждой стадией в среднем на 25%. К последней он составляет около 10%.

Остановимся на меланоме поподробнее, так как среди перечисленных новообразований, меланома является самой агрессивной. Пятилетняя выживаемость после удаления опухоли на I стадии составляет 98%. Но на последней стадии только 10% пациентов несмотря на современные технологии в состоянии пережить следующее пятилетие. Для России характерен высокий показатель роста выявления опухолей, более 5%. Самый

высокий показатель зарегистрирован в Самарской области [11]. Удаление Меланомы на ранних стадиях практически всегда приводит к выздоровлению.

Существующие методы диагностики, которые имеют высокую точность в основе своей являются инвазивными. К примеру исследование под микроскопом образцов ткани кожи после биопсии, которое позволяет установить гистологические характеристики при исследовании кожных патологий. Изначально стараются применять дерматологические методы исследования без нарушения целостности кожных покровов. Часто в сомнительных случаях, во избежание лишнего инвазивного вмешательства, пациента направляют на дерматологическую консультацию, после чего принимается решение о целесообразности биопсии. При этом гистология считается основным методом исследования кожных патологий при имеющихся поражениях, при явном подозрении на онкологическое заболевание или для определения типа имеющегося доброкачественного образования.

Но существующие методы диагностики рака кожи, относящиеся к неинвазивной диагностике, имеют невысокую точность, что в некоторых случаях может оказаться критично для пациента. Поэтому задача нахождение эффективных и точных методов диагностики заболевания является основной в современной онкологии. Использование современных, в том числе инструментальных методов уточняющей диагностики повышает точность предоперационного диагноза до 88% [12]. Проведенное исследование продемонстрировало, что 40% поражений кожи с подозрениями на злокачественность оказались доброкачественными опухолями [13].

Зная, что меланома кожи является очень прогрессивным и опасным заболеванием, необходим инструмент для диагностики, которое позволит с определенной точностью классифицировать данное заболевание. Одним из

методов неинвазивной диагностики является исследование оптических свойств тканей кожи с помощью комбинационного рассеяния.

Комбинационное рассеяние (КР) отслеживает изменение длины волны падающего света после обмена энергией кванта света с рассеивающими молекулами. Принцип работы КР заключается в том, что развитие кожных онкологических заболеваний сопровождается структурным на молекулярном уровне и химическом изменением ткани кожи. Все эти изменения регистрируются КР, которые уникальны для каждого молекулярного соединения. Сравнение таких спектров позволило классифицировать виды опухолей кожи [14].

## 1.2 Спектроскопия видимого и ультрафиолетового света, Раман-спектроскопия

Спектроскопическими методами анализа объектов называют методы взаимодействия анализирующего образца и электромагнитных излучений. Диапазон спектра волн продемонстрированных на рисунке 1.1 обычно бывает от радио волн до  $\gamma$ -излучения, другими словами от 0,1 см до  $10^{-11}$  м. С отдельно взятыми электромагнитными спектрами связаны различные способы анализа.

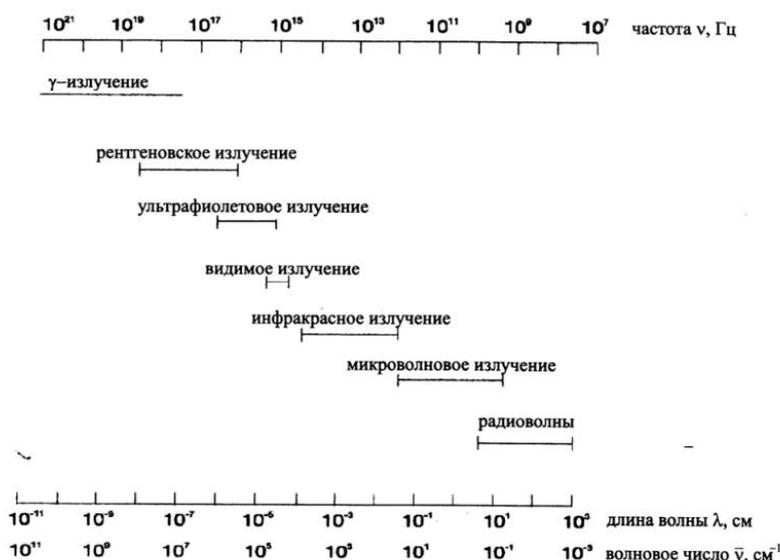


Рисунок 1.1 – Диапазон электромагнитного спектра

Спектр называется индивидуальной характеристикой вещества. Положение линий в спектре является мерой, с помощью которой проводят

анализ содержащегося вещества с некоторой точностью. Также на свойство вещества влияют поглощение света, перераспределение энергии в молекуле, образование первичных фотопродуктов и превращение их в первые химически устойчивые соединения.

Существует множество видов спектроскопии, но остановимся поподробнее на спектрах комбинационного рассеяния (КР) или Раман-спектроскопия.

Методы КР основаны на эффекте неупругого рассеяния, в результате которого происходит изменения частоты измененного рассеянного света. Причиной изменения частоты связано с переходом на более высокий колебательный уровень или добавление к электромагнитной волне энергии молекул для повышения или понижения колебательной частоты. Примеры спектров на рисунке 1.2.

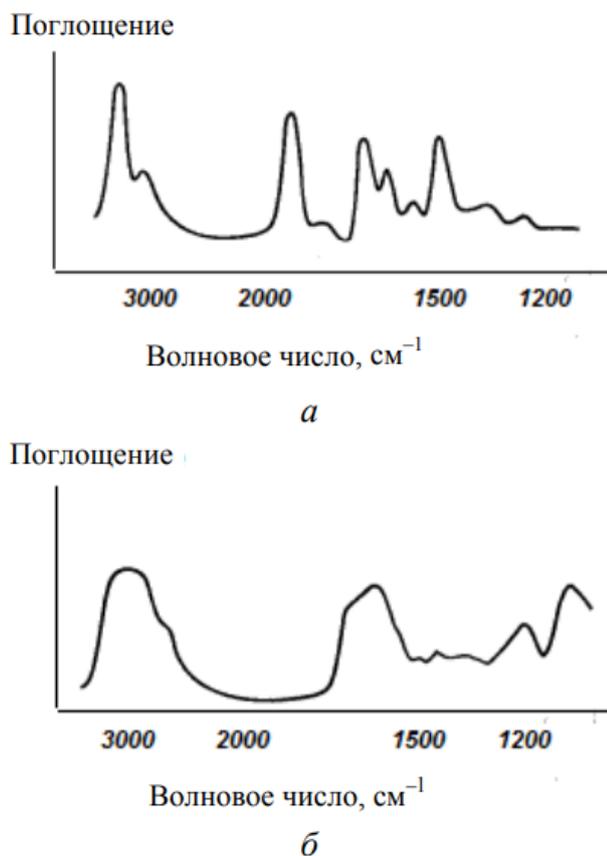


Рисунок 1.2 – ИК-спектры стеариновой кислоты (а) и ДНК (б)

В данной работе будет использоваться КР для регистрации патологий. Сбор данных произвел Самарский национальный исследовательский

университет имени академика С.П. Королева (СНИУ) и Самарский областной клинический онкологический диспансер (ГБУЗ СОКОД).

Схема аппаратуры для оптической спектроскопии изображена на рисунке 1.3 включает в себя источник излучения, отделение для пробы, диспергирующее устройство или монохроматор для разложения светового потока и систему детектирования и регистрации. В основном состоит из этого набора компонентов [16].



Рисунок 1.3 – Принципиальная схема оптического спектрометра.

Спектрометр комбинационного рассеяния по своей конструкции похож на спектрометр ультрафиолетовой области. Для наблюдения необходим источник интенсивного монографического излучения, а также обладающий высокой светосилой монохроматор и высокочувствительный детектор [16].

Регистрация спектров проводилась в диапазоне 780-1000 нм. Для каждого участка биоткани было зарегистрировано 3 независимых измерения. В данной работе были исследованы 700 спектров предварительно размеченных спектров кожи и опухолей и 34 злокачественных новообразований: 22 меланом, 12 базальноклеточного рака и 10 доброкачественных образований для классификации рака кожи. Все *in vivo* исследования проводились с согласия пациентов. Все пациенты были старше 18 лет. Данные были получены из СНИУ совместно с ГБУЗ СОКОД и являются их собственностью.

### **1.3 Общая характеристика искусственных нейронных сетей**

В 1943 году Уоррен Мак-Каллок предложил понятия искусственного нейрона и искусственной нейронной сети. Эти идеи, в 1958 году продолжил

нейрофизиолог Фрэнк Розенблатт. Он предложил схему устройства, моделирующего процесс человеческого восприятия, и назвал его «персептроном» (от латинского *perceptio* – восприятие).

Искусственный нейрон с  $n$  входами изображен на рисунке 1.4, представляет собой модель, похожую на настоящий нейрон. На входы  $i$  поступают  $x_i$  сигналы. Активационная функция  $\varphi$  в теле нейрона вычисляет значение на основе которого делается вывод возбуждать нейрон или нет. Функция активации выбирается из условия поставленной задачи. Входные сигналы имеют вес  $w_i$ , который меняется в соответствии формулы при умножении входного сигнала на веса и дальнейшей корректировки данных весов в случае некорректного ответа. Переданная в нейрон информация умножается на веса, суммируется и передается функции активации для получения выходного сигнала.

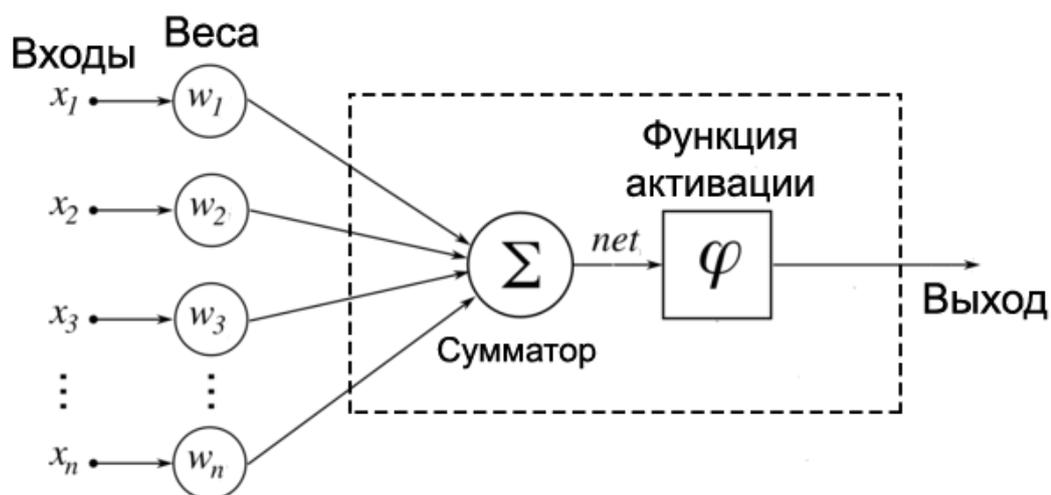


Рисунок 1.4 – Схема Искусственного нейрона

Если воспринимать каждый нейрон в нейронной сети как отдельный элемент со своей функцией активации, весом и определенным выходным сигналом, то набор из нескольких нейронов представляет собой композицию, называемой нейронной сетью. Существуют различные модели нейронных сетей, различающиеся в основном функциями активации, способом связи между слоями и временем передачи информации.

В большинстве случаев предполагается, что каждый нейрон вносит вклад в выходной нейрон, с которым он соединён. Полный вклад такой сети определяется формулой (1.1)

$$S_k f = f \sum_i w_{jk} t y_j t + \theta_k t , \quad (1.1)$$

где  $w_{jk}$  – вклад весов, который будет положительный или отрицательный,  $y_j$  – значение входных сигналов,  $\theta_k$  – вес фиктивного нейрона. Это самый просто пример суммирования весов, существуют более сложные.

На рисунке 1.5 изображены часто используемые функции: а) пороговая ступенчатая функция, б) сигмоидная функция активации, в) гиперболический тангенс.

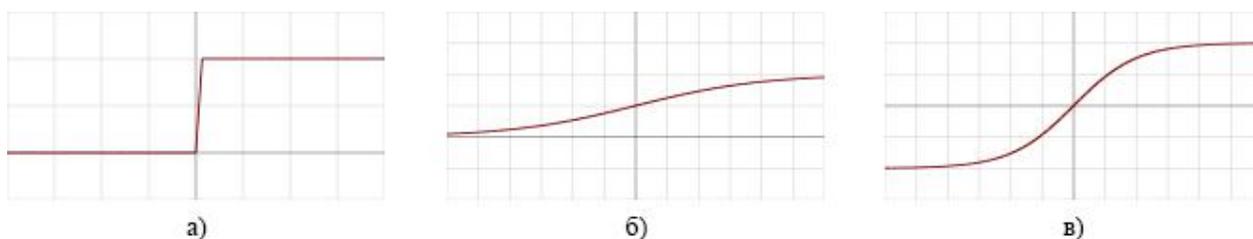


Рисунок 1.5 – Графики некоторых функций активации

Уравнение графика ступенчатой функции (1.2).

$$f(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases} \quad (1.2)$$

Уравнение функции сигмоида активации или логической функции (1.3). Данная активационная функция служит нелинейной усилительной характеристикой нейрона. Выражается наклонной кривой при определённом уровне возбуждения. Гроссберг обнаружил, что такой подход решает проблему шумового насыщения.

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

Уравнение гиперболического тангенса (1.4).

$$f(x) = th(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1.4)$$

Другая широко используемая активационная функция — это гиперболический тангенс.

## 2 ПРОЕКТИРОВАНИЕ НЕЙРОСЕТЕВЫХ АРХИТЕКТУР

### 2.1 Постановка задачи для дифференциации меланомы кожи

В прошлой главе были рассмотрены спектры КР и что они из себя представляют и как с их помощью можно определять структуру вещества на материальном уровне. Задача данной работы повысить точность диагностики меланомы среди патологий. Рассмотрим метод, с помощью которого возможен анализ спектров КР для определения патологий кожи и определение меланомы среди патологий.

На рисунке 2.1 продемонстрированы данные полученные из СНИУ

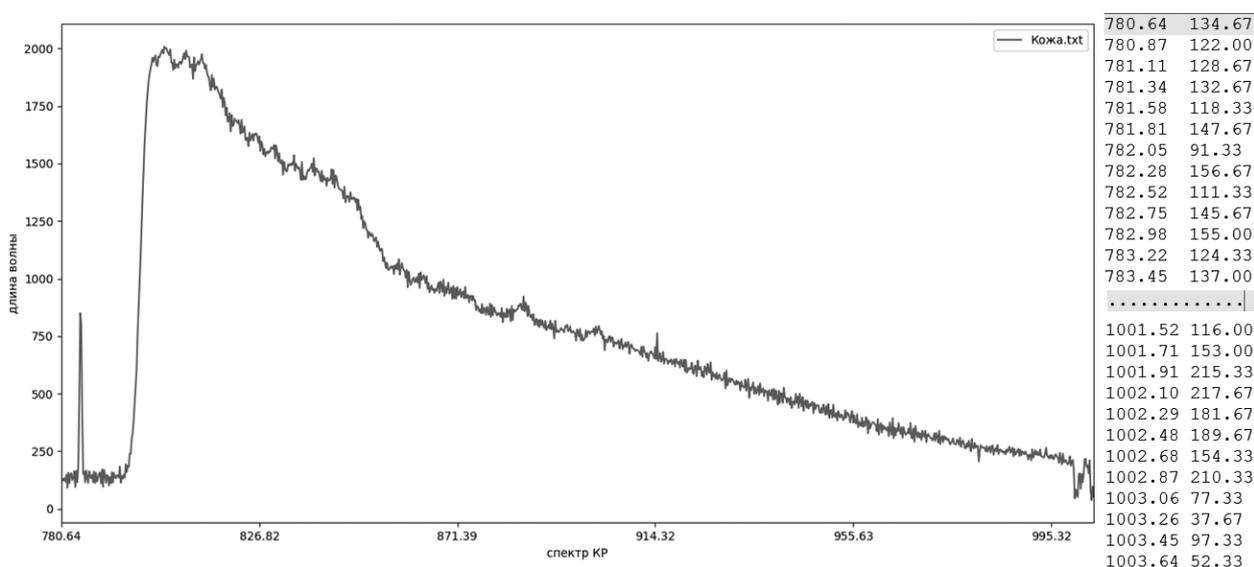


Рисунок 2.1 – Спектры кожи комбинационного рассеяние

Зная, какой спектр является кожей, а какой спектр является опухолью или меланомой, необходимо реализовать архитектуру, которая будет сама на неразмеченных данных определять свойство данного ей спектра и, используя данные о свойствах спектра, определять является ли полученный спектр кожи пациента меланомой или нет.

Также мы рассмотрели в первой главе, что такое нейроны и что они способны обучаться на основе настройки весовых коэффициентов для дальнейшего предсказания результатов.

В данной работе будет разработана архитектура, построенная из нескольких нейронов называемая нейронной сетью. Данная сеть произведет

обучение с учителем на спектрах КР для предсказания опухоли среди спектров и определения меланомы среди патологий.

## 2.2 Подготовка данных и обучение искусственной нейронной сети

Для начала с помощью скрипта нужно обработать полученные данные из СНИУ и перевести их из формата .txt в формат .xlsx. Для этого потребуется написать программу на языке Python. На рисунке 2.2 продемонстрировано из чего состоят данные КР, а именно из двух столбцов, длины волны и полученного спектра.

780.64	134.67
780.87	122.00
781.11	128.67
781.34	132.67
781.58	118.33
781.81	147.67
782.05	91.33
782.28	156.67
782.52	111.33
782.75	145.67
782.98	155.00
783.22	124.33
783.45	137.00
.....	
1001.52	116.00
1001.71	153.00
1001.91	215.33
1002.10	217.67
1002.29	181.67
1002.48	189.67
1002.68	154.33
1002.87	210.33
1003.06	77.33
1003.26	37.67
1003.45	97.33
1003.64	52.33

Рисунок 2.2 – Спектры КР в формате .txt

Данные были переведены в табличный вид, продемонстрированный на рисунке 2.3, и представляют из себя .xlsx файл с меткой на конце именуемой label, где К – это здоровая кожи, а О – это опухоль пациента. Видно часть данных таблицы, где каждая строка — это пациент и его полученные данные спектра от длины волны.

1002,68	1002,87	1003,06	1003,26	1003,45	1003,64	label
2915,67	2920,33	2664,33	2604,67	2683,67	2619,33	O
3047,67	3053,33	2662,00	2618,00	2684,00	2630,00	O
777,00	762,67	81,33	87,67	103,33	90,33	K
645,00	634,67	131,00	120,00	137,67	125,67	O
433,67	439,33	78,00	90,00	92,67	75,67	K
911,00	933,33	87,67	124,67	106,33	116,67	O
509,00	471,33	57,00	36,33	60,67	35,00	K
1462,33	1441,00	110,67	78,33	130,33	91,67	O
357,33	363,33	69,00	84,00	73,33	77,33	K
1038,00	1027,00	83,33	95,33	100,67	89,00	O
2729,67	2716,67	2605,33	2552,67	2610,33	2578,67	K
330,00	324,00	68,00	63,00	68,67	82,00	K
290,33	317,33	60,00	81,67	72,00	78,00	O
200,67	190,67	55,67	75,67	46,33	81,67	O

Рисунок 2.3 – Спектры КР в формате .xlsx

Эти данные потребуются для удобного обучение нейронных сетей.

### 2.3 Задача обучения искусственной нейронной сети

Процессом обучения искусственной нейронной сети (ИНС), как говорилось ранее, является подбор параметров, настройка архитектуры и подбор весовых коэффициентов для решения поставленной задачи. Задача данной работы заключается в нахождение патологий кожи на спектрах КР, которые представляют из себя таблицу спектров с предварительно размеченными данными.

На рисунке 2.4 продемонстрирована схема обучения ИНС с учителем. На схеме видно, что изменение весов происходит благодаря сравнению полученных ответов ИНС с реальными ответами в размеченных загруженных тренировочных данных. Сравнение ответов ИНС называется сигналом ошибки — это разность между желаемым сигналом и текущим откликом нейронной сети [15].

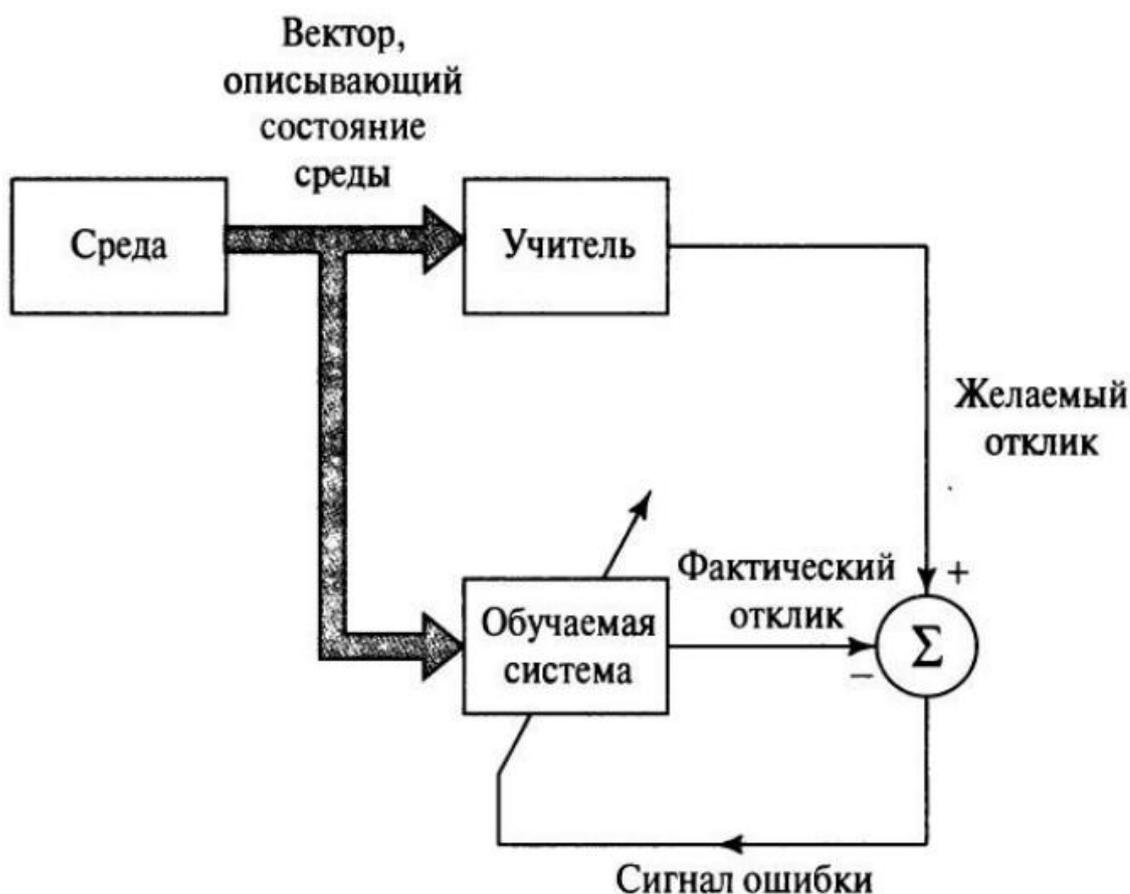


Рисунок 2.4 – Схема обучения сети с учителем.

Существуют несколько формулировок обучения ИНС, но в основном они строятся на двух формулировках. Основными характеристиками данных обучающих систем является: обучающее множество, контрольное множество и погрешность.

1. Обучающее множество:  $\{(x^i, d^i)\}, i = 1, \dots, k, x^i = (x_1^i, x_2^i, \dots, x_n^i)$  – это фактические входные сигналы, а  $d^i = (d_1^i, d_2^i, \dots, d_m^i)$  – это ожидаемые выходные сигналы. На данном множестве происходит обучение сети.

2. Контрольное множество:  $\{(x^i, d^i)\}, i = k + 1, \dots, k + s$  где,  $x^i = (x_1^i, x_2^i, \dots, x_n^i), d^i = (d_1^i, d_2^i, \dots, d_m^i)$ . На данном множестве происходит сравнение полученной обученной ИНС, и проверка точности его решения на данных которые она еще не видела.

3. Коэффициент погрешности  $\varepsilon$ , требуется для архитектур в которых требуется добиться определенной точности решения ИНС.

Далее приведены формулировки задачи обучения ИНС.

Дано: обучающее множество и погрешность  $\varepsilon_0 > 0$

Необходимо: привести весовые коэффициенты  $w_{ij}$  к виду, что  $F x^i - d^i < \varepsilon_0 \forall i = 1, \dots, k$

Дано: обучающее множество и контрольное множество, погрешность  $\varepsilon_0 > 0$  и  $\varepsilon_1 > 0$  для обучающего и контрольного множества соответственно.

Необходимо: привести весовые коэффициенты  $w_{ij}$  к виду, что  $F x^i - d^i < \varepsilon_0 \forall i = 1, \dots, k$  и  $F x^i - d^i < \varepsilon_1 \forall i = k + 1, \dots, k + s$

Примером задачи минимизации целевой функции  $E(w)$

Дано: обучающее множество и погрешность  $\varepsilon_0 > 0$

Необходимо: найти весовые коэффициенты  $w_{ij}^k$ , такие, что  $E w = \frac{1}{2} \sum_{i=1}^k (F x^i - d^i)^2 = \frac{1}{2} \sum_{i=1}^k (F x^i - d^i)^2 < \frac{\varepsilon_0^2}{2} k = \varepsilon$

В связи с тем, что функции активации гладкие, можно использовать другие методы оптимизации в том числе и градиентные методы.

ИНС должна настроить веса по полученным примерам, такой подход называется обучением с учителем. Обучение ИНС начинается с подачи сигналов во входной слой. Данный этап называется прямым ходом алгоритма обратного распространения. Во время прямого хода, каждый узел скрытого слоя получает от всех остальных узлов сигналы входного слоя, которые перемножаются на веса узлов и затем суммируются. Сигналы же являются преобразованными значениям функций активации. В выходном слое происходит аналогичное преобразование. Значения выходного слоя сравниваются с ожидаемыми значениями. Сравнение между полученными и ожидаемыми значениями называют ошибкой. Данную ошибку распространяют между скрытыми слоями и такое направление называют обратным ходом алгоритма обратного распространения ошибки.

Сети разделяют на два вида обучения это:

Детерминированный метод обучения — это метод, когда параметры сети меняются по итерациям и основываются на текущих входах. Примером

детерминированного метода обучения является метод обратного распространения ошибки. Другими словами, параметры сети меняются по итерациям и основываются на текущих входах.

Стохастический метод обучения — это метод использует случайный подбор параметров и сохраняет лучшие из них.

Рассмотрим, детерминированный метод или алгоритм обратного распространения ошибки, поподробнее. Это алгоритм использует стратегию подбора весов ИНС с применением градиентных методов оптимизации. Его основой является целевая функция (2.1), которая формируется в виде квадратичной суммы разностей для единичной выборки, полученных и ожидаемых значений выходных сигналов и функция (2.2) для обучающих выборок  $j = 1, 2, \dots, p$ .

$$E w = \frac{1}{2} \sum_{k=1}^M y_k - d_k^2 \quad (2.1)$$

$$E w = \frac{1}{2} \sum_{j=1}^p \sum_{k=1}^M y_k^j - d_k^j^2 \quad (2.2)$$

Обучение производится в несколько этапов:

1. Берется вектор обучающей выборки  $x = (x_1, x_2, \dots, x_N)$  и рассчитываются значения выходных сигналов. Далее определяется значение выходных сигналов  $y = (y_1, y_2, \dots, y_M)^T$ .

2. Минимизируется значения целевой функции. Применение градиентных методов оптимизации весов производится по формуле (2.3).

$$w_{k+1} = w_k + \Delta w, \quad (2.3)$$

где  $\Delta w = \eta p(w)$ ,  $\eta$  — это скорость обучения,  $p(w)$  — направление движения в многомерном пространстве весовых коэффициентов  $w$ .

Обучение ИНС требует определение вектора направления  $p(w)$  для слоя сети. Очевидное решение имеет только выходной слой сети. Для обучения внутренних слоев необходимо учитывать влияние этих слоев на итоговый выход сети и создаваемую ими погрешность. Рассмотрим

стратегию, созданную для решения этой проблем, алгоритм обратного распространения ошибки. Ранее рассматривалось, что веса нейронов корректируются в сторону убывания целевой функции, в сторону противоположной градиенту с координатами:

$$(\nabla_0 E, \nabla_1 E, \nabla_2 E, \dots, \nabla_N E)^T, \nabla_j E = \frac{\partial E}{\partial w_{ij}} \quad (2.4)$$

Компоненты градиента расписываются как:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_i} \cdot \frac{\partial y_i}{\partial u_i} \cdot y_i \cdot \frac{\partial u_i}{\partial w_{ij}} = y_i - d_i \cdot f' u_i \cdot x_j \quad (2.5)$$

Ее применение возможно только для выходного слоя. Для применения к внутреннему слою необходимо учитывать целевую функцию, рисунок 2.5.

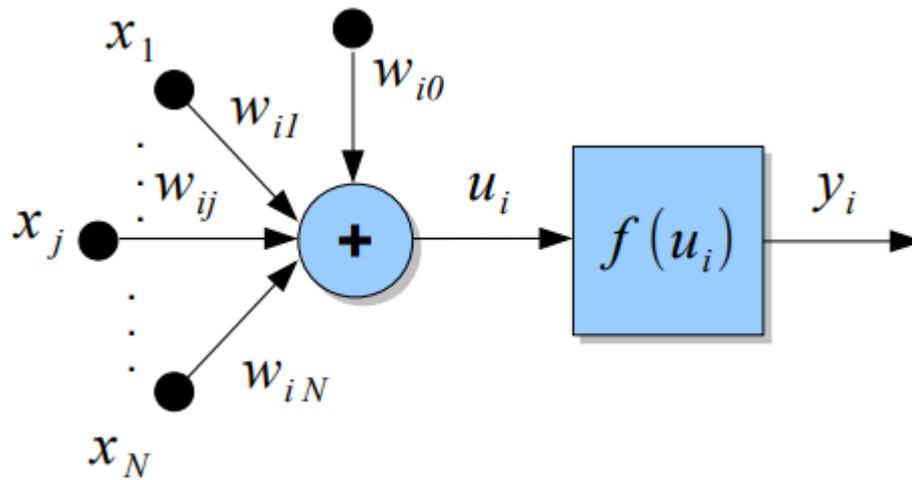


Рисунок 2.5 – Целевая функция нейронной сети

Рассмотрим выход нейрона  $j$  предпоследнего слоя  $n < N$  (2.6):

$$\frac{\partial E}{\partial y_i} = \sum_{k=1}^{K^{n+1}} \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial u_k} \cdot \frac{\partial u_k}{\partial y_j} = \sum_{k=1}^{K^{n+1}} \frac{\partial E}{\partial y_k} \cdot f' u_k \cdot x_{jk}^{n+1}, \quad (2.6)$$

где суммирование по всем  $K$  нейронам следующего слоя  $n + 1$  связано со следующим нейроном слоя  $n$ . Обозначим:

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot f' u_j = \sum_{k=1}^{K^{n+1}} \delta_k^{n+1} \cdot w_{jk}^{n+1} \cdot f' u_j = \delta_j^{n+1}. \quad (2.7)$$

При выходном слое  $N$ :  $\delta_j^{(N)} = (y_j^N - d_j) \cdot f'(u_j)$ . Таким образом для слоя  $n$  компоненты градиента вычисляются по формуле (2.8):

$$\Delta w_{ij}^{(n)} = -\eta \delta_j^n y_i^{n-1} \quad (2.8)$$

для первого слоя  $n = 1$ :  $y_i^{(1)} = x_i$

Для устранения эффекта осцилляции вводят коэффициенты инерционного обучения  $\Delta w_{ij}^n t + 1 = -\eta \delta_j^n y_i^{n-1} + \alpha w_{ij}^n t$ ,  $\alpha < 1$  или  $\Delta w_{ij}^n t + 1 = -\eta [\mu \Delta w_{ij}^n t + (1 - \mu) \delta_j^{(n)} y_i^{n-1}]$ ,  $\mu < 1$ .

Можно сделать вывод, что обратное распространение ошибки и другие рассмотренные методы и алгоритмы для обучения ИНС являются конечными факторами для успешной работы ИНС, но существуют проблемы, возникающие при работе ИНС, которые необходимо затронуть и решить. Архитектура многослойной искусственной нейронной сети состоит из расположенных на разных уровнях нейронов. В общем виде имеется входной слой и выходной слой, также возможно применение скрытых слоев, которые может насчитываться от одного и более слоев, количество нейронов в слое неограниченно и выбирается исходя из условия задачи.

На рисунке 2.6 показано строение такой сети, она имеет четыре слоя, где на входном слое  $N$  нейронов, на выходном слое  $K$  нейронов и один скрытый слой с выходными сигналами  $v_i$ .

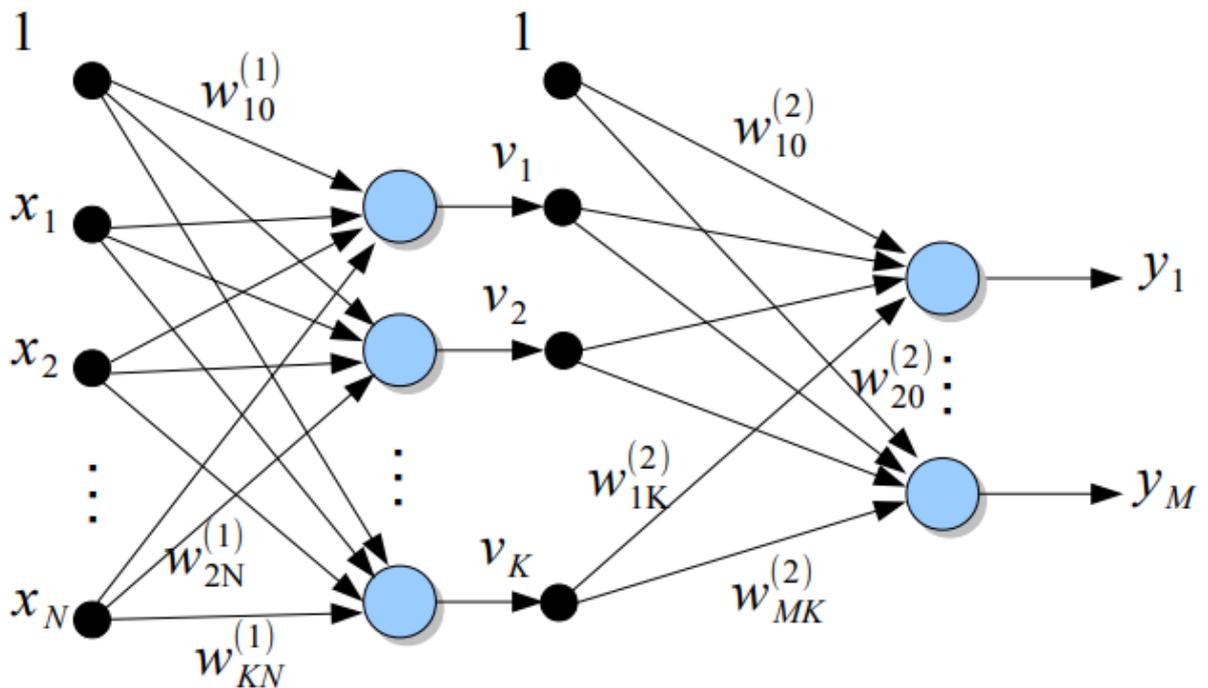


Рисунок 2.6 – Многослойная искусственная нейронная сеть

Многослойные ИНС состоят из множества искусственных нейронов и имеют систему обучения, связанную на настройке весов для всех слоев сети. На вход приходит сигнал  $x_N$ , который попадает на вход нейрона. Далее настраиваются веса  $w_{ij}$ , они попадают в формулу суммирования (1.1) и затем в функцию активации (1.2), (1.3), (1.4).

Настройка весов основывается на выходных сигналах и векторах фактических выходных сигналов.

Вектор фактических выходных сигналов:  $y = (y_1, y_2, \dots, y_M)^T$

Вектор ожидаемых выходных сигналов:  $d = (d_1, d_2, \dots, d_M)^T$

Цель обучения состоит в подборе весов  $w_{ij}^{(1)}$  и  $w_{ij}^{(2)}$  для слоев сети так, чтобы при заданных сигналах сеть выдавала с заданной точностью правильный ответ.

С учетом весов выходной сигнал  $i$  нейрона первого скрытого слоя равен  $v_i$  (2.9).

$$v_i = f \sum_{j=0}^N w_{ij}^1 x_j \quad (2.9)$$

При этом для входа на следующем слое принимается  $v_0 = 1$ . В выходном слое  $k$ -й нейрон вырабатывает выходной сигнал, вычисляемый по формуле (2.10):

$$y_k = f \sum_{i=0}^K w_{ki}^2 v_i = f \sum_{i=0}^K w_{ki}^2 \sum_{j=0}^N w_{ij}^1 x_j \quad (2.10)$$

Из этой формулы видно, что учитываются веса всех слоев.

#### 2.4 Проблемы обучения искусственной нейронной сети

Неопределенно долгий процесс обучения несет за собой множество проблем. Сложные задачи с большим количеством параметров могут требовать очень большое количество времени для обучения или ИНС может совсем не обучиться. Причиной этих проблем может являться паралич сети или попадание в локальный минимум.

Паралич в процессе обучения ИНС это проблем, когда веса могут стать очень большими, что приводит к тому что нейроны сети начинают работать при очень больших коэффициентах активационных функций и будет возвращать пропорциональные значения из-за чего процесс обучения может замереть. На данный момент проблема в теории плохо изучена. Проблему можно избежать, уменьшая размер шага, но это увеличивает время обучения. Любые методы восстановления после паралича на данном этапе являются экспериментальными.

Обратное распространения ошибки использует градиентный спуск для спуска вниз и корректировки весов в сторону минимума по поверхности ошибки. Поверхность ошибки в ИНС имеет большую разбросанность волн в пространстве высокой размерности. ИНС при таких раскладах может попасть в локальный минимум и из-за того, что все направленные идут вверх ИНС неспособна из нее выбраться. Рассмотренные статические методы могут избежать такие проблемы, но они очень медленные.

На рисунке 2.7 продемонстрированы две функции обработки ошибки, на оси абсцисс значения параметра, а по ординат – соответствующая функция ошибки. Они обе не имеют локальных минимумов, но, а) будет менее пригодной для градиентной оптимизации чем, б).

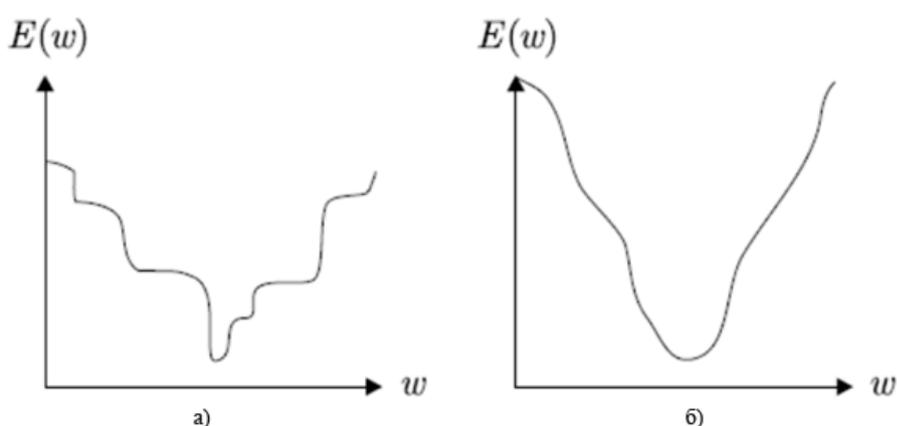


Рисунок 2.7 – Две функции ошибки одной размерности

Параметр обобщения показывает то, как хорошо ИНС работает на новых данных. Зная, что ИНС обучается на заранее известных данных, обобщение показывает эффективность модели на новых данных.

Математически, цель обучения ИНС это минимизация функции (2.11)

$$E_{true} = \int_{x,d} e f(x, w, d) p(d, x) dx dd, \quad (2.11)$$

где  $e$  – локальная функция стоимости,  $f$  – функция реализуемая с помощью ИНС,  $d$  – это ожидаемый,  $w$  – соответствующие веса в сети,  $p$  – вероятность распределения. Целью обучения это оптимизация весов  $w$ , или сведения (2.12) к минимуму:

$$w = \underset{x,d}{\operatorname{argmin}_w} \int_{x,d} e f(x, w, d) p(d, x) dx dd \quad (2.12)$$

$E_{true}$  – ожидаемая работа ИНС на новых данных выбранных из  $p(d, x)$ . Зная обучающие пары и функцию, которую надо минимизировать. Ее называют эмпирической функцией (2.13):

$$E = \frac{1}{N_p} \sum_{p=1}^{N_p} e(x_p, d_p) \quad (2.13)$$

Встает вопрос, как понять, что ИНС хорошо натренирована для минимизации (2.15). Рассмотрев обобщение и разобрав взаимодействие тренировочных данных и тестовых данных перейдем к проблеме, называемой переобучением.

Геометрическая интерпретация показывает, что при обучении ИНС процесс представляет собой разделение векторного пространства гиперплоскостью. Можно сделать вывод, что чем больше нейронов и параметров, тем сложнее поверхность на которой происходит разделение. Но здесь и появляется проблема. Обучив ИНС и прогнав по тренировочным данным, на выходе получается точное решение соответствующие ответам, но на тестовых данных точность решение резко снижается, что может быть связано с переобучением. Происходит это обычно с зашумленными данными.

На рисунке 2.8 показана аппроксимация некоторой функции  $y = f(x)$ , красная линия построена по точкам обучающей выборки и проходя через все точки имеет нулевую ошибку.

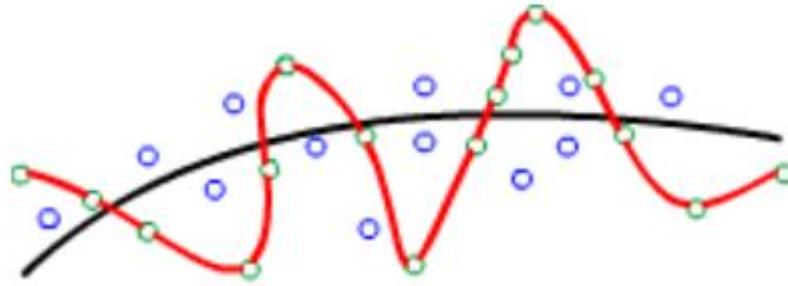


Рисунок 2.8 – Аппроксимация данных некоторой функцией

Но на тестовых данных, это синие точки, лучше работает черная гладкая линия. Можно сделать вывод, что уменьшение или увеличение числа нейронов влияет на работу ИНС и нужно грамотно делать подбор количества нейронов при архитектуре ИНС.

## 2.5 Альтернативные методы обратного распространения ошибки

Ранее рассматривался стохастический метод градиентного спуска, но это не единственный способ для оптимизации обратного распространения ошибки. Рассмотрим другие алгоритмы:

Adagrad – это более продвинутая техника обучения (относительно SGD), которая выполняет градиентный спуск с переменной скоростью обучения. Весам узлов, которые имели большие градиенты, дают большие градиенты, в то время как весам узлов, которые имели малые градиенты, дают небольшие градиенты.

Adagrad по сути является SGD со встроенным в алгоритм планировщиком скорости обучения для каждого узла. Таким образом, Adagrad улучшает SGD, предоставляя веса скоростей обучения, вместо того, чтобы довольствоваться единой скоростью обучения для всех узлов.

Чтобы лучше понять свойства Adagrad, нужно взглянуть на его уравнение (2.14):

$$\theta_{t+1} = \theta_t - \frac{\eta}{G_t + \epsilon} \cdot g_t, \quad (2.14)$$

где  $\theta_t$  – вес узла для генерации  $t$ ,  $G_t$  – диагональная матрица, содержащая квадраты всех предыдущих градиентов;  $\epsilon$  – очень маленький порядок  $e^{-8}$  члена регуляризации, который обеспечивает числовую стабильность,

предотвращая деление на 0;  $g_t$  – вектор градиентов для текущего поколения; и  $\eta$  – скорость обучения.

Однако недостаток Adagrad заключается и в матрице. Каждое обновление, которое хотим выполнить, будет положительным, и поэтому матрица знаменателя всегда будет расти по величине. При наличии достаточного количества поколений оно сократит скорость обучения и практически сведет на нет фактическое обучение.

Adadelta – это адаптация Adagrad, которая использует импульсные методы для решения монотонно уменьшающейся проблемы скорости обучения. Adadelta имеет каждое обновление градиента по каждому весу, которое является взвешенной суммой текущего градиента и экспоненциально убывающим средним из ограниченного числа прошлых обновлений градиента. Поскольку знаменатель градиента больше не монотонно увеличивается, скорость обучения более стабильна, то алгоритм в целом более устойчив.

RMSprop – это исправление для Adagrad, которое было предложено независимо от Adadelta. Он эквивалентен Adadelta, с одним отличием: скорость обучения дополнительно делится на экспоненциально убывающее среднее значение всех квадратов градиентов, например, глобальное значение настройки.

Adam выступает за адаптивную оценку момента. В дополнение к хранению экспоненциально убывающего среднего значения градиентов прошлого квадрата, таких как Adadelta и RMSprop, Адам также сохраняет экспоненциально убывающее среднее значение градиентов прошлого, аналогично импульсу.

Математически Adam начинает с вычисления оценок среднего значения и дисперсии (первого и второго моментов) градиента по следующей формулам (2.15) и (2.16):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad (2.15)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \quad (2.16)$$

где  $m_t$  – экспоненциально убывающее среднее значение прошлых градиентов,  $v_t$  – экспоненциально убывающее среднее значение прошлых квадратов градиентов.

Проблема использования этих значений в правиле обновления заключается в том, что они смещены в сторону 0, поскольку  $m$  и  $v$  инициализируются как нулевые векторы при первом запуске алгоритма. Поэтому Adam вводит еще одну поправку в свои формулы (2.17) и (2.18):

$$m_t = \frac{m_t}{1 - \beta_1} \quad (2.17)$$

$$v_t = \frac{v_t}{1 - \beta_2} \quad (2.18)$$

Включаем эти скорректированные значения, использующие импульс, в ту же формулу адаптивной скорости обучения, которую используют RMSprop (2.19):

$$\theta_{t+1} = \theta_t - \frac{\eta}{v_t + \epsilon} \cdot m_t \quad (2.19)$$

Поиск пути алгоритма Adam, по сути, состоит из шара с импульсом и трением. Adam смещает путь, по которому алгоритм идет к плоским минимумам на поверхности ошибки, замедляя обучение при движении по большому градиенту. AdaMax, как RMSprop, использует экспоненциально убывающую средневзвешенную оценку дисперсии градиента в своей формулировке. Дисперсия эквивалентна второму моменту или норме градиента. Норма определяется как  $L_n = \sqrt[n]{g^n}$

Бесконечная норма численно устойчива, потому что имеет асимптотически сходящееся поведение.

AdaMax более устойчив к градиентному шуму при обновлении, чем Adam, и имеет лучшую числовую стабильность. Именно поэтому это требует дальнейшего исследования

Nadam – это Adam, но с импульсом Нестерова вместо обычного импульса. Преимущество использования импульса Нестерова вместо обычного импульса такое же, как и в случае с SGD.

## **2.6 Разработка архитектуры и программная реализация искусственной нейронной сети**

Разобравшись в прошлой главе, как работает ИНС и с какими проблемами придется столкнуться, перейдем непосредственно к реализации архитектуры ИНС для классификации патологий рака кожи на спектрах КР. При выполнении бакалаврской работы использовался высокоуровневый язык программирования Python 3.6, библиотеками Keras и Scikit-learn, предоставляющий широкий инструментарий для решения задач машинного обучения.

Python — это высокоуровневый язык программирования общего назначения. Python поддерживает структурное, объектно-ориентированное, функциональное программирование. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

Keras — это библиотека нейронных сетей с открытым исходным кодом, написанная на Python. Он может работать поверх TensorFlow, Microsoft Cognitive Toolkit, Theano или PlaidML. Разработанный для быстрого экспериментирования с глубокими нейронными сетями, он ориентирован на удобство использования, модульность и расширяемость.

Scikit-learn — это библиотека машинного обучения программному обеспечению для языка программирования Python. Имеет различные алгоритмы классификации, регрессии и кластеризации, включая машины опорных векторов, случайные леса, повышение градиента, k-средних и DBSCAN, и предназначен для взаимодействия с числовыми и научными библиотеками Python NumPy и SciPy.

При создании архитектуры ИНС возникает проблема указания количества нейронов в слоях сети. Ранее говорилось, что ИНС имеет входной слой, слой на который передаются входные сигналы, его размерность определяется от размера входного сигнала. Этот слой не выполняет никаких вычислительных операций. Выходной слой схож подходом с входным слоем.

Количество нейронов выходного слоя определяется размерностью выходного сигнала.

Для выбора количества нейронов и слоев для входного и выходного слоя существуют несколько правил, но основной вопрос стоит в скрытых слоях. Если задача линейна разделима, то никаких скрытых слоев не потребуется. Проблема состоит в грамотном выборе количества слоев и влияния их на производительность.

Для определения числа нейронов в скрытых слоях можно воспользоваться эмпирически обоснованными правилами. Одним из таких правил для трехслойной сети является правило геометрической пирамиды (2.20):

$$p = \overline{nm}, \quad (2.20)$$

где  $p$  – число нейронов в скрытом слое,  $n$  – число нейронов во входном слое,  $m$  – число нейронов в выходном слое.

Для четырехслойной ИНС вычисляется по формуле (2.21):

$$r = \sqrt[3]{\frac{\overline{n}}{m}}, p_1 = mr^2, p_2 = mr, \quad (2.21)$$

где  $p_1$  – число нейронов в первом скрытом слое,  $p_2$  – число нейронов во втором скрытом слое.

Существуют множество других способов определения количества нейронов в скрытых слоях ИНС, которые основаны на расчетах, начинаемых с одного скрытого слоя и подбору нейронов в нем для поиска оптимального решения с заданной точностью.

Перед проектированием архитектуры необходимо поставить задачу, которую будет решать данная ИНС. Первую задачу, которую нужно решить, определить является ли полученный спектр кожей или опухолью, если данный образец окажется опухолью, то нужно классифицировать это новообразование.

Для решения данной проблемы воспользуемся многослойной ИНС. Рассмотрим сеть из трех слоев и четырех с двумя выходами и тремя

выходами. На рисунке 2.9 продемонстрирована ИНС состоящая из трех слоев и двумя выходами для бинарной классификации кожи и опухолей.

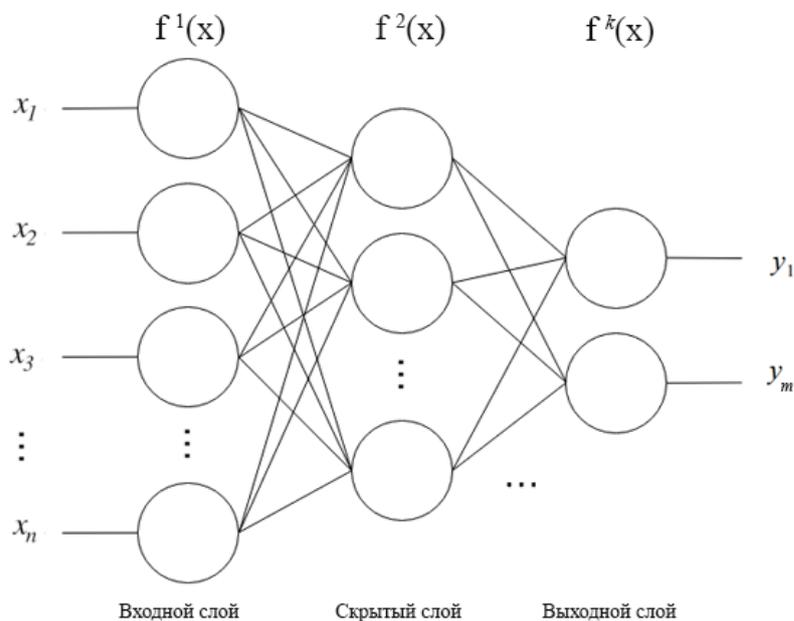


Рисунок 2.9 – Архитектура многослойной нейронной сети

Продемонстрирована многослойная ИНС, где  $x_n$  – это входной сигнал,  $y_m$  – это выходной сигнал,  $f^k(x)$  – это функция активации. Рассмотрим два варианта ИНС для бинарной классификации патологий:

1) Трехслойная ИНС, где один входной слой имеет  $n = 1044$  нейрона равный количеству входных классов, выходной слой  $m = 1$  нейрон равный бинарной классификации и один скрытый слой имеет по формуле (2.20) при  $n = 1044, m = 1, p = \sqrt{1 \cdot 1044} \approx 32$  нейрона,  $f^k(x)$  будет выбираться наилучшая с помощью подбора параметров между формулами (1.2), (1.3), (1.4).

2) Четырехслойная ИНС, где один входной слой имеет  $n = 1044$  нейрона равный количеству входных классов, выходной слой  $m = 1$  равный бинарной классификации и два скрытых слоя имеют по формуле (2.21) при  $n = 1044, m = 1, r = \sqrt[3]{\frac{1044}{1}} \approx 10$ ,  $p_1 = 100, p_2 = 10$  для двух скрытых слоев,  $f^k(x)$  будет выбираться наилучшая с помощью подбора параметров между формулами (1.2), (1.3), (1.4).

Следующие ИНС будут использоваться для классификации полученных опухолей:

3) Трехслойная ИНС, где один входной слой имеет  $n = 1044$  нейрона равный количеству входных классов, выходной слой  $m = 3$  нейрона равный количеству классификаций и один скрытый слой имеет по формуле (2.20) при  $n = 1044, m = 3, p = \sqrt[3]{3 \cdot 1044} \approx 56$  нейронов,  $f^k(x)$  будет выбираться наилучшая с помощью подбора параметров между формулами (1.2), (1.3), (1.4).

4) Четырехслойная ИНС, где один входной слой имеет  $n = 1044$  нейрона равный количеству входных классов, выходной слой  $m = 3$  нейрона равный количеству классификаций и два скрытых слоя имеют по формуле (2.21) при  $n = 1044, m = 3, r = \sqrt[3]{\frac{1044}{3}} \approx 7, p_1 = 147, p_2 = 21$  для двух скрытых слоев,  $f^k(x)$  будет выбираться наилучшая с помощью подбора параметров между формулами (1.2), (1.3), (1.4).

Для реализации архитектуры ИНС воспользуемся Python и библиотекой Keras, листинг кода предоставлен на рисунке 2.10.

```
model = Sequential()
model.add(Dense(1044, input_dim=train_dataset.shape[1],
                kernel_initializer='uniform', activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(32, kernel_initializer='uniform', activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(1, kernel_initializer='uniform', activation='sigmoid'))
model.compile(loss='binary_crossentropy',
              optimizer='rmsprop', metrics=['accuracy'])
```

Рисунок 2.10 – Листинг архитектуры трехслойной нейронной сети

Для создания архитектуры модели необходимо определить *model*, затем в модель добавляется входной слой размером 1044 нейрона, *input\_dim* равен количеству входных параметров, *kernel\_initializer* – это способ задания начальных весов в слое, *activation* – используемая функция активации в слое. В листинге видно, что сеть состоит из трех слоев, количество нейронов в слое определено и посчитано заранее и описано выше. Модель собирается в

*compile* и использует функцию потерь *loss* и оценку оптимизации *optimizer*. Слой Dropout необходим для установки параметров, при которых сеть не будет переобучаться.

На рисунке 2.11 продемонстрирована вывод в консоль работы ИНС.

```
epoch =    0 loss = 0.691378 acc = 52.07%
epoch =   100 loss = 0.592032 acc = 67.39%
epoch =   200 loss = 0.571561 acc = 68.88%
epoch =   300 loss = 0.564628 acc = 70.17%
epoch =   400 loss = 0.555430 acc = 70.24%
epoch =   500 loss = 0.552093 acc = 70.37%
epoch =   600 loss = 0.552909 acc = 69.83%
epoch =   700 loss = 0.545722 acc = 71.53%
epoch =   800 loss = 0.545182 acc = 71.59%
epoch =   900 loss = 0.531145 acc = 72.34%
```

```
Loss, accuracy on test data:
0.5967 74.86%
```

Рисунок 2.11 – Вывод в консоль работы трехслойной нейронной сети

Значения *epoch* равняется периоду эпохи обучения, *loss* скалярное значение, которое необходимо минимизировать во время нашего обучения модели. Чем ниже потеря, тем ближе наши прогнозы к истинным ярлыкам, *accuracy* демонстрирует точность работы. Последняя строка демонстрирует работу на тестовых данных на которых ИНС не обучалась и используя свои веса пыталась предсказать ответ.

Воспользуемся библиотекой Scikit-learn для автоматизации подбора параметров, а именно функцией GridSearchCV.

Листинг кода подбора параметров представлен на рисунке 2.12.

```

# create model
model = KerasClassifier(build_fn=create_model, verbose=0)
# define the grid search parameters

_batch_size = [10, 20, 40, 60, 80, 100]
_epochs = [500, 1000, 5000]
_learn_rate = [0.001, 0.01, 0.1, 0.2, 0.3]
_momentum = [0.0, 0.2, 0.4, 0.6, 0.8, 0.9]
_init_mode = ['uniform', 'lecun_uniform', 'normal', 'zero',
              'glorot_normal', 'glorot_uniform', 'he_normal', 'he_uniform']
_activation = ['softmax', 'softplus', 'softsign',
               'relu', 'tanh', 'sigmoid', 'hard_sigmoid', 'linear']
_dropout_rate = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]

param_grid = dict(batch_size = _batch_size,
                  epochs = _epochs,
                  learn_rate = _learn_rate,
                  momentum = _momentum,
                  optimizer=_optimizer,
                  init_mode=_init_mode,
                  activation=_activation,
                  dropout_rate=_dropout_rate)
grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=1)
grid_result = grid.fit(X, Y)

```

Рисунок 2.12 – Листинг функции подбора параметров для нейронной сети

Для этого составим массив интересующих нас параметров, на основе которых будет происходить поиск по сетке, таких как *batch\_size* – это гиперпараметр градиентного спуска, который контролирует количество обучающих выборок, которые необходимо обработать до обновления внутренних параметров модели, *epochs* – это гиперпараметр градиентного спуска, который контролирует количество полных проходов через набор обучающих данных, *learn\_rate* скорость обучения стохастического градиентного спуска (SGD), *momentum* – это параметр, который ускоряет SGD в соответствующем направлении и демпфирует колебания., *init\_mode* определяют способ задания начальных случайных весов слоев, *activation* – используемая функция активации в слое, *dropout\_rate* – значение между 0 и 1 для остановки переобучения.

Перекрестную проверку будем реализовывать с помощью библиотеки Scikit-learn и функцию StratifiedKFold. Пример работы перекрестной проверки представлен в листинге на рисунке 2.13.

```

kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=seed)
cvscores = []
for train, test in kfold.split(X, Y):
    # create model
    model = Sequential()
    model.add(Dense(1044, input_dim=train_dataset.shape[1],
                    kernel_initializer='uniform', activation='relu'))
    model.add(Dropout(0.25))
    model.add(Dense(32, kernel_initializer='uniform', activation='relu'))
    model.add(Dropout(0.25))
    model.add(Dense(3, kernel_initializer='uniform', activation='sigmoid'))
    model.compile(loss='sparse_categorical_crossentropy',
                  optimizer='adam', metrics=['accuracy'])

    # Fit the model
    model.fit(X[train], Y[train], epochs=1000, batch_size=100, verbose=0)
    # evaluate the model
    scores = model.evaluate(X[test], Y[test], verbose=0)
    print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
    cvscores.append(scores[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (numpy.mean(cvscores), numpy.std(cvscores)))

```

Рисунок 2.13 – Листинг функции перекрестной проверки для нейронной сети  
 Результат работы функции продемонстрирована на рисунке 2.14.

```

acc: 75.00%
acc: 75.00%
acc: 25.00%
acc: 50.00%
acc: 75.00%
acc: 100.00%
acc: 66.67%
acc: 33.33%
acc: 100.00%
acc: 66.67%
66.67% (+/- 23.57%)

```

Рисунок 2.14 – Вывод в консоль работы функции перекрестной проверки

Значение асс означает ассигасу, другими словами точность решение ИНС. Можно сделать вывод, что перекрестная проверка дает более точное представление точности обученной сети.

## 2.7 Перекрестная проверка и подбор параметров по сетке

Перекрестная проверка, иногда называемая оценкой ротации, или тестирование вне выборки – это любой из различных методов проверки модели для оценки того, как результаты статистического анализа будут обобщаться в независимый набор данных [18]. В основном используется в условиях, когда целью является прогнозирование насколько точно

прогностическая модель будет работать на практике. В задаче прогнозирования модели обычно дают набор данных с известными ответами, по которым проводится обучение, и набор данных по которым проверяется модель. Цель перекрестной проверки состоит в том, чтобы проверить способность модели прогнозировать новые данные, которые не использовались при ее оценке, чтобы обозначить такие проблемы, как переобучение или систематическая ошибка выбора, и дать представление о том, как модель будет обобщена для независимый набор данных [19].

На рисунке 2.15 продемонстрирована примерная работа перекрестной проверки. Один раунд перекрестной проверки включает разделение выборки данных на дополнительные подмножества, выполнение анализа на одном подмножестве, называемом обучающим набором и проверку правильности анализа на другом подмножестве, называемом проверочным набором или набором тестирования. Чтобы уменьшить изменчивость, в большинстве способов выполняется несколько раундов перекрестной проверки с использованием разных разделов, и результаты проверки объединяются или усредняются по раундам, чтобы дать оценку прогнозирующей эффективности модели.

Перекрестная проверка – это метод для оценки эффективности модели. Это также полезно при определении гиперпараметров модели, в том смысле, что эти параметры приведут к наименьшей ошибке теста. С помощью Scikit-Learn можно начать работу со всеми видами техник валидации.

Таким образом, перекрестная проверка сочетает средние показатели пригодности в прогнозировании, чтобы получить более точную оценку эффективности прогнозирования модели [20].

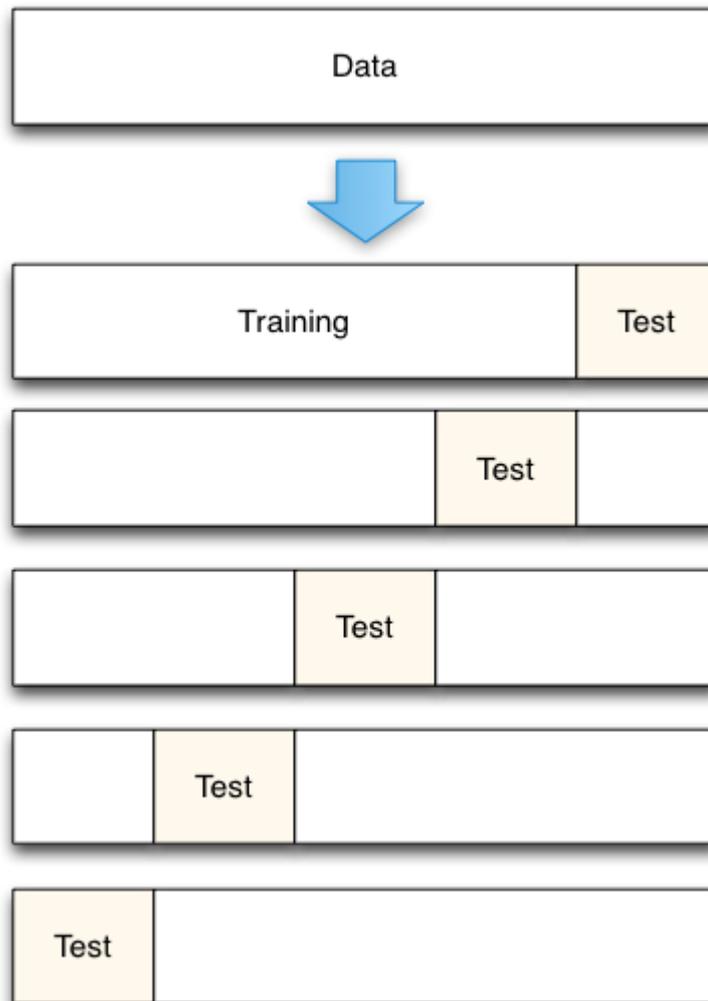


Рисунок 2.15 – Перекрестная проверка

В машинном обучении оптимизация или настройка гиперпараметра - это проблема выбора набора оптимальных гиперпараметров для алгоритма обучения. Гиперпараметр - это параметр, значение которого используется для управления процессом обучения. Напротив, значения других параметров запоминаются.

Поиск по сетке – это традиционным способом выполнения гиперпараметрической оптимизации, которая представляет собой просто исчерпывающий поиск в заданном вручную подмножестве пространства гиперпараметров алгоритма обучения. Алгоритм поиска в сетке должен руководствоваться некоторой метрикой производительности, обычно измеряемой путем перекрестной проверки на обучающем наборе или оценки на удерживаемом проверочном наборе [21] [22].

Поскольку пространство параметров может включать в себя действительные или неограниченные пространства значений для определенных параметров, перед применением поиска по сетке могут потребоваться ручные установки границ и дискретизации.

В таблицах 2.1, 2.2 продемонстрированы подборы параметров и результаты тестирования подбора параметров для первой трехслойной ИНС. Значения в ячейках демонстрируют точность работы сети на тестовых данных. Для остальных сетей сделаем аналогичные подборы параметров.

Таблица 2.1 –Параметры обучения для трёхслойной нейронной сети

Количество эпох обучения и обучающей выборки				
Количество эпох обучения	Количество обучающих выборок			
	10	20	50	100
1000	0,650	0,707	0,623	0,665
2000	0,702	0,685	0,645	0,660
5000	0,677	0,692	0,675	0,670
Параметры градиентного спуска				
Скорость обучения	Параметр ускорения			
	0,0	0,2	0,4	0,6
0,001	0,485	0,492	0,492	0,485
0,01	0,635	0,647	0,655	0,655
0,1	0,592	0,652	0,648	0,663
0,2	0,637	0,562	0,613	0,552

Изучая таблицу, можно сказать, что при большом количестве эпох ИНС не получает выгоды в пользу улучшения работы сети и малое значения обучающей выборки не дает сети попасть в паралич, но влияет на скорость работы сети. Можно сделать вывод, что наилучшую точность показывает связка, где количество эпох равно 1000, а размер выборки 20. Будем использовать данные параметры для дальнейшего обучения. Из таблицы 2.2 видно, что стохастический градиентный спуск показывает результаты хуже, чем другие алгоритмы оптимизации ошибки.

Таблица 2.2 –Параметры обучения для трёхслойной нейронной сети

Другие алгоритмы оптимизации				
Алгоритм оптимизации	Точность оптимизации			
RMSprop	0,613			
Adagrad	0,635			
Adadelat	0,700			
Adam	0,618			
Adamax	0,712			
Nadam	0,703			
Функции активации				
Слой искусственной нейронной сети	Функция активации			
	linear	relu	tanh	softmax
Первый	0,707	0,722	0,718	0,682
Второй	0,657	0,710	0,715	0,728
Третий	0,174	0,570	0,467	0,71
Параметр остановки переобучения				
Слой	Параметр остановки переобучения			
	0,1	0,2	0,3	0,4
Между первым и вторым	0,692	0,7121	0,707	0,7204
Между вторым и третьим	0,687	0,683	0,705	0,685

По результатам подбора, сделаем вывод, что оптимальными параметрами для сети будут: алгоритм оптимизации Adamax, функция активации первого слоя Relu, второго слоя Sigmoid, третьего слоя Sigmoid, параметр остановки обучения между первым и вторым равен 0,4 и между вторым и третьим равен 0,3. Для остальных нейронных сетей произведем аналогичный подбор параметров.

## 2.8 Программа принятия решения и ее блок-схема

Обучив ИНС на спектрах КР воспользуемся программой принятия решений, которая изображена в блок-схеме на рисунке 2.16. Для написания программы принятия решений потребуется сохранить обученные нейронные сети.

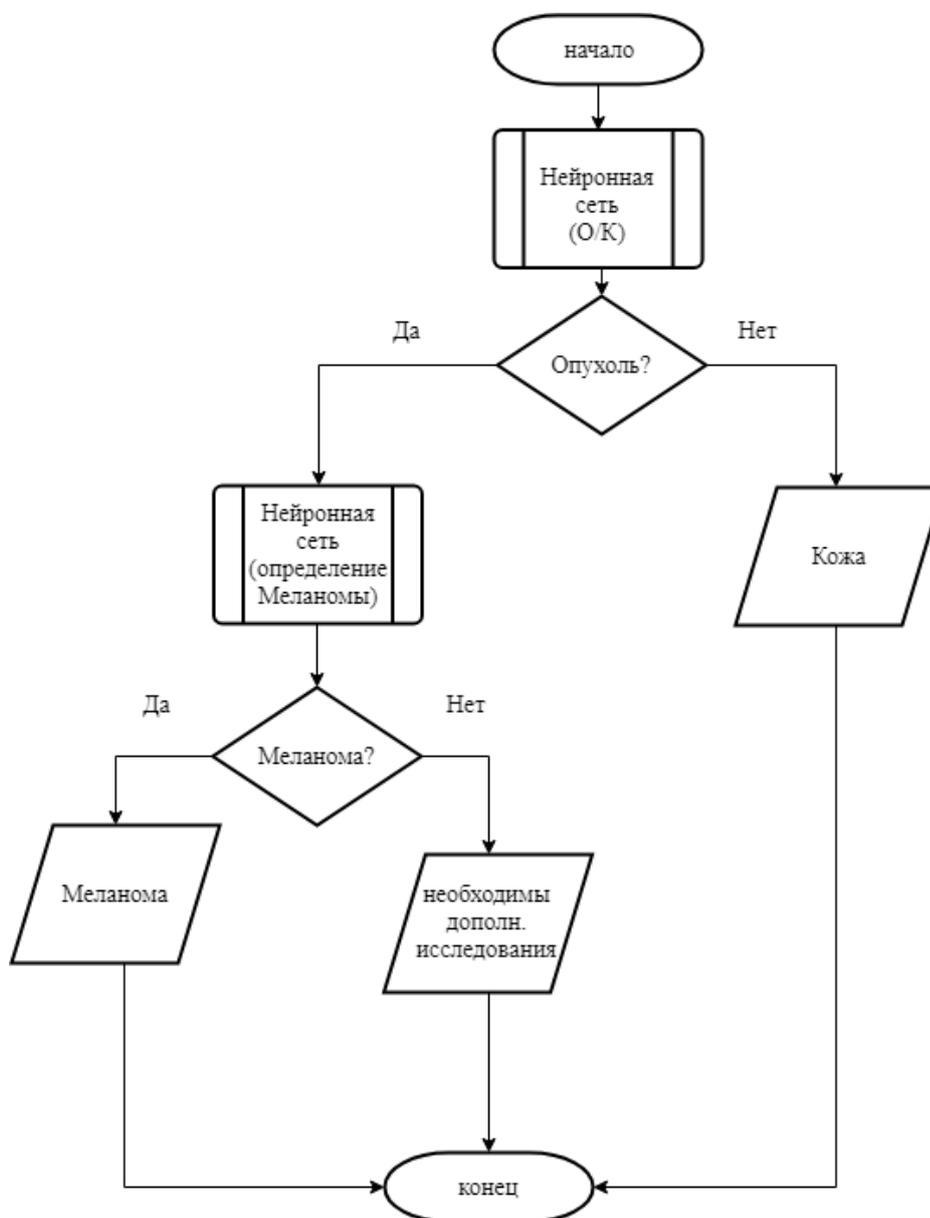


Рисунок 2.16 – Блок-схема программы

С помощью программы написанной на Python, листинг которой прикреплен в приложении А, автоматизируем вывод результатов определения является ли спектр кожи КР пациента кожей или опухолью и если является опухолью, то ИНС даст ответ доброкачественная это или злокачественная опухоль и является ли она меланомой.

## 3 ТЕСТИРОВАНИЕ НЕЙРОСЕТЕВЫХ АРХИТЕКТУР

### 3.1 Разделение данных и их нормализация

С использованием скрипта, написанного на Python, разделим данные на два массива значений: данные и ответы. На рисунке 3.1 продемонстрирован пример вывода консоли, в котором видно, что первый массив показывает данные спектров, а второй, под чертой, классификация кожи, где 0 – это кожа, 1 – это опухоль.

```
[[2873.  2843.  2899.  ... 2614.67 2657.67 2633.33]
 [ 219.33  210.33  238.  ...   78.33  130.33  91.67]
 [ 134.    77.33  134.67 ...   23.    88.33  37.  ]
 ...
 [2783.33 2758.67 2809.  ... 2567.  2632.33 2572.33]
 [ 138.67  190.67  143.33 ...   62.67  104.  76.33]
 [2870.33 2827.33 2905.  ... 2604.67 2683.67 2619.33]]
-----
[1 1 0 ... 1 0 0]
```

Рисунок 3.1 – Вывод данных в консоль

В раннее рассмотренной библиотеки Scikit-learn воспользуемся функцией нормализации данных. Нормализация - это метод, который часто применяется как часть подготовки данных для обучения ИНС. Целью нормализации является изменение значений числовых столбцов в наборе данных на общую шкалу без искажения различий в диапазонах значений. Для обучения ИНС каждый набор данных не требует нормализации. Требуется только тогда, когда объекты имеют разные диапазоны. Хорошее эмпирическое правило заключается в том, что входные переменные должны иметь небольшие значения, вероятно, в диапазоне от 0 до 1 или стандартизироваться с нулевым средним и стандартным отклонением, равным единице. На рисунке 3.2 продемонстрированы нормализованные данные.

```

[[[0.00227585 0.00255762 0.00248545 ... 0.00091749 0.00157511 0.00132216]
 [0.00263468 0.00291389 0.00292098 ... 0.00066581 0.0022193 0.00112393]
 [0.02793179 0.02744398 0.02799481 ... 0.0257615 0.02605021 0.02587101]
 ...
 [0.02209238 0.02193082 0.02235201 ... 0.02049505 0.02093476 0.02048711]
 [0.00151883 0.00183941 0.0014582 ... 0.00065948 0.00108916 0.00067165]
 [0.02096683 0.02055164 0.02118883 ... 0.01895629 0.01943418 0.01904318]]]

```

Рисунок 3.2 – Нормализованные данные спектров КР

Далее воспользуемся функцией для разделения данных на тренировочные и тестовые в отношении 80% и 20%. Это необходимо для определения точности работы ИНС на данных, которая она не видела.

В данной работе будут использованы сырые данные, которые не будут подвержены предобработке. Данные не подвергаются обработки, например, сглаживанию, так неизвестно как это повлияет на конечный результат. Нормализация данных не изменяет данные, а приводит их к общему стандарту, с которым программе быстрее и проще работать.

### 3.2 Методы оценки точности работы искусственной нейронной сети

В медицине для определения точности диагностических методов используют параметры, которые называют характеристиками исследования. А именно, чувствительность и специфичность.

Формула чувствительности (3.1) определяет доли положительных результатов среди всех проведенных тестов:

$$\frac{TP}{TP + FN} \cdot 100\%, \quad (3.1)$$

где  $TP$  – это положительные результаты,  $FN$ – это ложноотрицательные результаты.

Формула специфичности (3.2) определяет при отсутствии заболеваний ложноположительных результатов как доля истинно отрицательных результатов среди здоровых лиц в группе исследуемых:

$$\frac{TN}{TN + FP} \cdot 100\%, \quad (3.2)$$

где  $TN$  – это отрицательные результаты,  $FP$  – это ложноположительные результаты [23].

Далее с помощью (3.1) и (3.2) определим точность метода диагностики с использованием ИНС.

### 3.3 Тестирование искусственных нейронных сетей и программы принятия решений

На основе спроектированных и обученных ИНС, используя программу принятия решений, проведем тестирование на данных, которые модели не видели. Были разработаны четыре архитектуры:

1. Трехслойная ИНС для распознавания кожи или охули.
2. Четырехслойная ИНС для распознавания кожи или охули.
3. Трехслойная ИНС для распознавания меланомы.
4. Четырехслойная ИНС для распознавания меланомы.

Тесты проводились на подготовленных данных, в которых присутствуют четыре вида спектров. Спектры кожи, меланомы, базалиомы и доброкачественной опухоли, например, невус. Данные спектры ИНС не видела и, используя настроенные веса, будет предсказывать результаты. Программа принятия решений для начала определит является ли спектр опухолью или кожей, а затем определит какой именно видом опухоли является взятый спектр с определенной точностью.

Трехслойная ИНС для распознавания кожи и опухоли продемонстрировала точность работы в 72% на тренировочных данных и 74% на тестовых, перекрестная проверка показала средний показатель среди 10 выборок в 71% точности и ошибка обратного распространения показала средний параметр в 0,5. Сохраним веса и модель для программы принятия решения и протестируем на подготовленных данных. На рисунке 3.3 продемонстрирован вывод в консоль программы.

```
0 Опухоль точность = 0.9545033
1 Опухоль точность = 0.96836793
2 Необходимы дополнительные исследования
3 Опухоль точность = 0.6489959
4 Кожа точность = 0.6322997212409973
5 Кожа точность = 0.8682993650436401
6 Кожа точность = 0.7519520670175552
```

Рисунок 3.3 – Вывод в консоль программы для трехслойной нейронной сети

Из полученных результатов можно сделать вывод, что первые две опухоли с 95% и 96% вероятностями являются опухолями, что соответствует действительности потому, что первый и второй образец являлись меланомой и базалиомой соответственно, но второй тест оказался недостаточно точным, так как не смог превысить установленный порог точности.

Далее потребуется определить к какому виду опухолей относятся полученные результаты. На рисунке 3.4 видны полученные результаты классификации опухолей. В тесте участвовали 4 спектра меланомы кожи, 1 базалиома, 1 доброкачественная опухоль и 2 здоровые кожи. Нейронная сеть с 95% вероятностью определила, что спектр является меланомой и угадала результаты всех 4 случаев. Можно сделать вывод, что необходимо больше данных о других патологиях для обучения сети так как спектров меланомы было больше половины.

```
0 Опухоль точность = 0.8515589
0 это Меланома 0.68467474
1 Опухоль точность = 0.935591
1 Необходимы дополнительные исследования
2 Опухоль точность = 0.5496582
2 это Меланома 0.9989588
3 Опухоль точность = 0.9904008
3 это Меланома 0.82368207
4 Кожа точность = 0.7319290637969971
5 Кожа точность = 0.5296484529972076
6 Опухоль точность = 0.9389075
6 это Меланома 0.8864779
7 Опухоль точность = 0.566931
7 это Меланома 0.99896216
8 Опухоль точность = 0.98134804
8 это Меланома 0.8684043
```

Рисунок 3.4 – Вывод в консоль классификации опухоли трехслойной нейронной сети

В таблице 3.1 продемонстрированы результаты работы всех 4 архитектур. Точность сети на перекрестной проверки дает более общую картину чем проверка на первичной тестовой выборке.

Таблица 3.1 – Результаты точности классификаций нейронных сетей

Вид архитектуры сети	Точность работы сети на тестовой выборке	Точность сети после перекрестной проверки
Трёхслойная сеть определения кожи или опухоли	74,17%	71.01%
Трёхслойная сеть определения меланомы	62,50%	79,17%
Четырёхслойная сеть определения кожи или опухоли	68,21%	70,21%
Четырёхслойная сеть определения меланомы	62.50%	75,00%

Воспользуемся формулами (3.1) и (3.2) для расчёта чувствительности и специфичности и получим результаты, продемонстрированные в таблицах 3.2 и 3.3.

Таблица 3.2 – Результаты чувствительности

Вид архитектуры сети	Трёхслойная сеть определения меланомы	Четырёхслойная сеть определения меланомы
Трёхслойная сеть определения кожи или опухоли	70%	80%
Четырёхслойная сеть определения кожи или опухоли	60%	80%

Таблица 3.3 – Результаты специфичности

Вид архитектуры сети	Трёхслойная сеть определения меланомы	Четырёхслойная сеть определения меланомы
Трёхслойная сеть определения кожи или опухоли	30%	20%
Четырёхслойная сеть определения кожи или опухоли	40%	20%

В результате лучше всего продемонстрировали работу трёхслойная сеть для определения опухоли среди спектров с точностью 71%, и четырёхслойная сеть для определения меланомы с точностью 80%, результаты вполне можно назвать удовлетворительными, но для других опухолей потребуется больше исследований и данных для обучений.

## ЗАКЛЮЧЕНИЕ

Ранняя диагностика рака кожи является актуальной проблемой на данный момент и чтоб повысить качество обследований в данной работе выпускной квалификационной работе решалась проблема повышения точности диагностики рака кожи путем разработки системы, работающей со спектрами комбинационного рассеяния кожи с применением искусственных нейронных сетей. Они обладают высокой эффективностью и находят широкое применение в различных областях, в том числе и в медицине.

Основными результатами работы являются:

1. Проведен анализ литературы связаны с раком кожи, способами диагностики, спектрам комбинационного рассеяния и искусственным нейронным сетям.
2. Рассмотрены проблемы обучения искусственной нейронной сети.
3. Были рассмотрены, спроектированы архитектуры искусственных нейронных сетей.
4. Реализованы нерестовые архитектуры на языке Python с использованием библиотек Keras и Scikit-learn. Подобраны оптимальные параметры и обучены нейронные сети на заранее подготовленных данных.
5. Тестирование показало, что алгоритма с 70% точностью определяет патологию от не патологий и с 80% определяет меланому кожу среди спектров КР.

В результате данной исследовательской работы была разработана архитектура для дифференциации меланомы кожи, который использует неинвазивный метод распознавания на спектрах КР при помощи нейронных сетей. Также планируются дальнейшие исследования для доработки алгоритма и улучшения его работы.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Зайцев В. Ф., Жидков С. А., Корик В. Е. Рак кожи: Методические рекомендации. – Минск: БГМУ, 2007.
2. Кисличко А. Г., Попов М. Ю., Григорович М. С. Опухоли кожи. Руководства для врачей общей практики. – М., 2015.
3. LeCun Y., Boser B., Denker J., Henderson D., Howard R., Hubbard W., Jackel L. Backpropagation Applied to Handwritten Zip Code Recognition // Neural Computation. – Vol. 1. – 1989. – Pp. 541-551.
4. H A Haenssle C Fink R Schneiderbauer F Toberer T Buhl A Blum A Kalloo A Ben Hadj Hassen L Thomas A Enk L Uhlmann Reader study level-I and level-II Groups Annals of Oncology, Volume 29, Issue 8, August 2018, Pages 1836–1842
5. Каприна А.Д., Старинского В.В., Петровой Г.В. Состояние онкологической помощи населению России в 2017 году. – М.: МНИОИ им. П.А. Герцена – филиал ФГБУ «НМИЦ радиологии» Минздрава России, 2018. – илл. - 236 с.
6. А.И. Юрченко, Н.И. Индилова, Н.Н. Потекаев Неинвазивные методы диагностики базальноклеточного рака кожи // Вестник Национального медико-хирургического Центра им. Н.И. Пирогова. - 2010. - С. 124-127.
7. Шляхтунов Е. А., Гидранович А. В., Луд Н. Г., Луд Л. Н., Кожар В. Л., Прокошин А. В. Рак кожи: современное состояние проблемы // Вестник Витебского государственного медицинского университета. - 2014. - С. 20-28.
8. Ганцев Ш.Х. , Юсупов А.С. Плоскоклеточный рак кожи // Практическая онкология. - 2012. - №2. - С. 80-87.
9. Зайцев, В. Ф. Рак кожи: метод. рекомендации / В. Ф. Зайцев, С. А. Жидков, В. Е. Корик. – Минск: БГМУ, 2007. – 18 с.
10. Козлов С.В., Захаров В.П., Моряттов А.А., Братченко И.А., Артемьев Д.Н. Возможности спектроскопии комбинационного рассеяния для

дифференциальной диагностики новообразований кожи // Конференция «Современные проблемы онкологии и гематологии». - 2015. - С. 542-546.

11. Козлов, С.В. Особенности заболеваемости меланомой кожи в самарской области // С.В. Козлов, Е.Ю. Неретин // Евразийский онкологический журнал. – 2014. – № 1. – С. 114.

12. Lin, M.J. Diagnostic accuracy of malignant melanoma according to subtype / M.J. Lin, V. Mar, C. McLean, R. Wolfe, J.W. Kelly // Australasian Journal of Dermatology. – 2012. – Vol. 55(1). – P. 35-42.

13. Hallock, G.G. Prospective study of the accuracy of the surgeon's diagnosis in 2000 excised skin tumors / G.G. Hallock, D.A Lutz // Plast Reconstr Surg. – 1998. – Vol. 101. – P. 1255–61.

14. Козлов, С.В. Возможности спектроскопии комбинационного рассеяния для дифференциальной диагностики новообразований кожи / С. В. Козлов, В.П. Захаров, А.А. Морятов, И.А. Братченко, Д.Н. Артемьев // Известия Самарского научного центра Российской академии наук. Социальные, гуманитарные, медико-биологические науки. – 2015. – Т. 17, № 2-3. – С. 542-547.

15. Хайкин, С. Нейронные сети: полный курс, 2-е изд., испр. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2006. — 1104 с.

16. М. Отто Современные методы аналитической химии 3-е издание Москва: Техносфера, 2008. - 544 с.

17. Ежов А., Шумский С., Нейрокомпьютинг и его применение в экономике и бизнесе, 1998

18. Geisser, Seymour (1993). Predictive Inference. New York, NY: Chapman and Hall. ISBN 978-0-412-03471-8.

19. Cawley, Gavin C.; Talbot, Nicola L. C. (2010). "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation" (PDF). 11. Journal of Machine Learning Research: 2079–2107.

20. Grossman, Robert; Seni, Giovanni; Elder, John; Agarwal, Nitin; Liu, Huan (2010). "Ensemble Methods in Data Mining: Improving Accuracy Through

Combining Predictions". Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool. 2: 1–126.

21. Chin-Wei Hsu, Chih-Chung Chang and Chih-Jen Lin (2010). A practical guide to support vector classification. Technical Report, National Taiwan University.

22. Chicco D (December 2017). "Ten quick tips for machine learning in computational biology". BioData Mining. 10 (35): 35. doi:10.1186/s13040-017-0155-3. PMC 5721660. PMID 29234465.

23. Королюк И.П. Медицинская информатика : Учебник / - 2 изд., перераб. и доп. - Самара : ООО «Офорт» : ГБОУ ВПО «СамГМУ».2012.- 244 с. 2012

## ПРИЛОЖЕНИЕ А

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
from keras.models import model_from_json
from keras.optimizers import SGD

import warnings
warnings.filterwarnings("ignore")

def CreateData(dataFrame):
    dataset = np.array( dataFrame.iloc[:, 0:1044])
    target = dataFrame.loc[:,['label']]
    target = np.array(target.replace(to_replace=['D', 'M', 'B'], value=[0, 1, 2]))
    target = target.astype('int32')
    target.transpose()
    x=np.where(np.isnan(dataset))[0]
    x=np.unique(x)
    i=0
    for item in x:
        dataset=np.delete(dataset, item-i, axis=0)
        target=np.delete(target, item-i)
        i+=1
    return target, dataset

_learn_rate = 0.1
_momentum = 0.6
_optimizer_1 = SGD(lr=_learn_rate, momentum=_momentum)
_optimizer_2 = 'Adamax'

path="test_class_raman.xlsx"
path=r"test_class_raman.xlsx"

dataFrame = pd.read_excel(path)

test_target, test_dataset = CreateData(dataFrame)

normalized_test_dataset = preprocessing.normalize(test_dataset)

json_file_model_1 = open("b_2_classifier_4_layer_Main_vkr.json","r") ## bin
model
load_model_json_1 = json_file_model_1.read()
json_file_model_1.close()
```

```

json_file_model_2 = open("m_2_classifier_4_layer_Main_vkr.json","r") ## class
model
load_model_json_2 = json_file_model_2.read()
json_file_model_2.close()

loaded_model_1 = model_from_json(load_model_json_1)
loaded_model_2 = model_from_json(load_model_json_2)

loaded_model_1.load_weights("b_2_classifier_4_layer_Main_vkr_weights.h5")
loaded_model_1.compile(loss='binary_crossentropy', optimizer=_optimizer_2,
metrics=['accuracy'])
#scores_1 = loaded_model_1.evaluate(normalized_test_dataset, test_target,
verbose=0)
#print("Точность работы на тестовых данных: %.2f%%" % (scores_1[1]*100))
prediction_1 = loaded_model_1.predict(normalized_test_dataset)

loaded_model_2.load_weights("m_2_classifier_4_layer_Main_vkr_weights.h5")
loaded_model_2.compile(loss='sparse_categorical_crossentropy',
optimizer=_optimizer_2, metrics=['accuracy'])
#scores_2 = loaded_model_2.evaluate(normalized_test_dataset, test_target,
verbose=0)
#print("Точность работы на тестовых данных: %.2f%%" % (scores_2[1]*100))
prediction_2 = loaded_model_2.predict(normalized_test_dataset)

#print(prediction_1)
#print(prediction_2)

for i in range(len(normalized_test_dataset)):
    elm = -1
    for j in range(len(prediction_1[i])):
        if prediction_1[i][j] >= 0.5:
            elm = 0
        elif prediction_1[i][j] <= 0.5:
            elm = 1
    if elm == 0:
        print(i, "Опухоль", 'точность =', prediction_1[i][j])
        x = -1
        elm = -1
        for j in range(len(prediction_2[i])):
            if x <= prediction_2[i][j]:
                x = prediction_2[i][j]
                elm = j
        if elm == 1 and x >= 0.8:
            print(i, "это Меланома", x)
        else:

```

```
print(i, "Необходимы дополнительные исследования")

elif elm == 1:
    print(i, "Кожа", 'точность =', 1 - prediction_1[i][j])
else:
    print(i, "Необходимы дополнительные исследования")
```