Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «**Информатика и системы управления**»

КАФЕДРА «**Программное обеспечение ЭВМ и информационные технологии**»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

НА ТЕМУ:

«Метод персонализации пользователей социальных сетей для целевой рекламы с использованием нейронных сетей»

Студент группы **ИУ7-85Б**	_____	**Д.О. Склифасовский**
	(Подпись, дата)	(И.О. Фамилия)
Руководитель ВКР	_____	**К.Л. Тассов**
	(Подпись, дата)	(И.О. Фамилия)
Нормоконтролер	_____	**Д.Ю. Мальцева**
	(Подпись, дата)	(И.О. Фамилия)

***2022** г.*

РЕФЕРАТ

Расчетно-пояснительная записка 55 с., 15 рис., 0 табл., 19 ист., 1 приложение, 20 листов презентационного материала.

Цель работы – разработать программное обеспечение для персонализации пользователей социальных сетей для целевой рекламы с использованием нейронных сетей.

Для достижения поставленной цели необходимо решить следующие задачи:

1) провести анализ существующих методов персонализации пользователей социальных сетей;
2) изучить способы получения информации о пользователях социальных сетей;
3) разработать метод персонализации пользователей социальных сетей на основе модифицированной нейронной сети;
4) разработать программное обеспечение, реализующее этот метод;
5) провести исследование применимости разработанного программного обеспечения.

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Нейронная сеть [1] – это совокупность клеток, которые связаны между собой синапсами. Нейронная сеть берет начало от общечеловеческой нервной системы. Ввиду этого программистами заложено то, что компьютер имеет возможность анализировать и сортировать огромное количество информации.

Машинное обучение [2] – нахождение отображения, в частности алгоритма классификации, который строится по множеству, называемому обучаемой выборкой, а качество обучения проверяется по множеству, называемому тестовой выборкой.

Нейрон [3][с. 6] – элемент, который вычисляет выходной сигнал (по определенному правилу) из совокупности входных сигналов.

Нейросеть [4] – это нейронная сеть, другими словами, это некая ткань, которая состоит из определенных клеток – нейронов, или нервных клеток. Данная структура выполняет громадную работу по восприятию, анализу информации из окружающей среды и реакции на разного рода раздражители.

Обучение нейронной сети [5, с. 31-32] – это процесс определения весов соединений между нейронами таким образом, чтобы сеть приближала необходимую функцию с заданной точностью.

Кластеризация [6] – это объединение объектов в группы (кластеры) на основе схожести признаков для объектов одной группы и отличий между группами.

Кластеризация [7] – метод нахождения кластерной структуры в наборе данных, который характеризуется наибольшим различием между различными кластерами.

Перцептрон [8] – математическая или компьютерная модель восприятия информации мозгом (кибернетическая модель мозга), предложенная Фрэнком Розенблаттом.

ВВЕДЕНИЕ

На данный момент на долю социальных сетей приходится треть всей рекламы в интернете.

Целевая реклама и персонализация занимают прочные позиции в области продвижения различных товаров. Данная отрасль развивается быстрыми темпами и важно успеть отслеживать изменения в данной сфере.

Исследования в этой области привлекли значительное внимание по двум основным причинам.

Во-первых, объем информации о товаре, доступной клиентам, постоянно растет, и поэтому желательно помочь клиентам разобраться в этой информации, чтобы найти наиболее подходящий им продукт или услугу.

Во-вторых, понимание потребностей текущих и потенциальных клиентов является неотъемлемой частью управления взаимоотношениями с клиентами.

Возможность точного, а также эффективного определения потребности клиентов и, в результате, выдачи им рекламы товаров, которые они сочтут желательными, открывает огромные возможности для роста бизнеса.

Применение нейронной сети в маркетинговой деятельности позволит выдавать наиболее подходящие товары, рекламные продукты, услуги непосредственному клиенту, что позволит повысит эффективность методов стимулирования сбыта и будет являться фактором устойчивого функционирования предприятия на рынке в условиях жесткой конкурентной борьбы, неопределенности и влияния значительных внешних факторов на его деятельность.

Цель данной работы – разработка программного обеспечения для персонализации пользователей социальных сетей для целевой рекламы с использованием нейронных сетей.

Для достижения поставленной цели необходимо решить следующие задачи:

1) провести анализ существующих методов персонализации пользовате-

лей социальных сетей;

2) изучить способы получения информации о пользователях социальных сетей;
3) разработать метод персонализации пользователей социальных сетей на основе модифицированной нейронной сети;
4) разработать программное обеспечение, реализующее этот метод;
5) провести исследование применимости разработанного программного обеспечения.

1 Аналитический раздел

В этой части рассматривается предметная область, выделяется объект исследования, и выполняется обзор существующих методов решения проблемы, поставленной в работе, либо методы решения смежных проблем.

1.1 Постановка задачи

Рассмотрим задачу персонализации пользователей социальных сетей для целевой рекламы с использованием нейронных сетей. На вход алгоритма персонализации подается индентификатор пользователя социальной сети. Выходом алгоритма является номер кластера, или же индекс целевой рекламы, которая интересна пользователю. Таким образом, задача персонализации пользователя делится на 2 этапа: получения доступной информации о пользователе из социальной сети и определение к какому кластеру пользователь относится. Постановка задачи представлена на рисунке 1.

Рисунок 1 – Постановка задачи

1.2 Обзор существующих решений

Сегодня существует множество хороших решений для рекомендательных систем для многих бизнес-областей. У большинства социальных сетей есть собственные алгоритмы персонализации пользователей для предоставления рекламы различных товаров или услуг.

1.2.1 Facebook

У Facebook самая сложная модель ранжирования. Она учитывает множество различных факторов. Техническая реализация находится в закрытом до-

ступе. Решение состоит из алгоритмов и методов, каждый из которых позволяет учитывать конкретный атрибут или явление в среде, решать определенную проблему. Facebook развивает систему итеративно и регулярно публикует обновление списка атрибутов, влияющих на ранжирование и основные принципы. Итеративность и постоянное развитие — неотъемлемые элементы хорошей рекомендательной системы.

1.2.2 ВКонтакте

Социальная сеть ВКонтакте использует сервис под названием «Церебро Таргет» для мониторинга открытых действий своих пользователей, сбора и систематизации полученных данных. Этот сервис используется для получения портрета необходимой целевой аудитории и определения места ее сосредоточения. Данный сервис является платным.

1.2.3 Yandex Zen

Дзен лента – это интеллектуальная алгоритмическая программа, которая анализирует публикуемый писателем материал и рекомендует его читателям в соответствии с их интересами. Таким образом информация распространяется по определенным направлениям.

Яндекс.Дзен способен предоставлять пользователям персонализированный контент. Платформа использует алгоритмы машинного обучения, чтобы рекомендовать новости пользователям на основе их предыдущего взаимодействий с площадкой или поисковой системой. Данные алгоритмы находятся в закрытом доступе.

1.3 Выбор социальной сети для получения информации о пользователях

Для формирования обучающей выборки необходимо выбрать наиболее подходящую социальную сеть, в которой зарегистрировано множество пользователей, а также такую, которая предоставляет данные о своих пользователях в открытом доступе.

1.3.1 Социальная сеть ВКонтакте

Социальной сетью ВКонтакте пользуется более 50 млн человек в день. Она является крупнейшей социальной сетью в России и странах СНГ. Основным преимуществом данной сети является открытое и бесплатное API, которое позволяет собирать большое количество информации о пользователях социальных сетей. Также аудитория ВКонтакте является в большей части русскоязычной.

1.3.2 Социальная сеть Facebook

Социальная сеть Facebook появилась в 2004 году как внутренняя социальная сеть в Гарвардском университете. Несмотря на то, что он начинал как способ поддержания связи, он стал удобным инструментом бизнеса, который мог точно нацеливаться на аудиторию и доставлять рекламу людям, которые, скорее всего, захотят их продукты или услуги.

По данным компании, ежедневно в Facebook заходят 1,28 млрд человек.

Также Facebook имеет открытое API для доступа к большому количеству информации.

Основной проблемой является то, что 4 марта 2022 году было принято решение о блокировке доступа к сети на территории Российской Федерации.

1.3.3 Социальная сеть Twitter

Twitter - американский сервис блогов и социальная сеть, в которой пользователи публикуют сообщения и взаимодействуют с ними. Пользователи взаимодействуют с «Твиттером» через браузер, мобильное приложение или через API.

Проблемой данной социальной сети является то, что нельзя собрать много информации о пользователях, а также с 4 марта 2022 года социальная сеть Twitter заблокирована на территории Российской Федерации.

1.3.4 Вывод

В качестве выбранной социальной сети для получения информации о людях была выбрана сеть ВКонтакте, так как у нее есть открытое API, большое количество информации о пользователях, а самое главное – она доступна на территории Российской Федерации.

1.4 Алгоритмы решения задач кластеризации

Кластеризация (или кластерный анализ) [9] – это задача разбиения множества объектов на группы, называемые кластерами.

Главное отличие кластеризации от классификации состоит в том, что перечень групп четко не задан и определяется в процессе работы алгоритма.

1.4.1 K-means++

Алгоритм k-средних является наиболее известным и используемым методом кластеризации.

Кластеризация k-means широко изучалась с различными расширениями в литературе и применялась в различных существенных областях [10].

Данный метод разбивает множество элементов векторного пространства на заранее известное число кластеров. Алгоритм стремится минимизировать среднеквадратичное отклонение на точках каждого кластера.

Основная идея данного алгоритма заключается в том, что на каждой итерации заново вычисляется центр масс для каждого кластера, полученного на последнем шаге, затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Алгоритм завершается, когда на какой-то итерации не происходит изменения кластеров.

Интуитивно алгоритм инициализации использует тот факт, что хорошая кластеризация относительно распределена, поэтому при выборе нового центра кластера предпочтение следует отдавать тем, которые находятся дальше от ранее выбранных центров.

Основная идея модифицированного K-means++ состоит в том, чтобы вы-

бирать центры один за другим контролируемым образом, где текущий набор выбранных центров будет стохастически смещать выбор следующего центра [11].

Вычисление цетроида:

$$\text{centroid}(Y) = \frac{1}{|Y|} \sum_{y \in Y} y \tag{1}$$

Необходимо определить стоимость Y по отношению к C как:

$$\phi_Y(C) = \sum_{y \in Y} d^2(y, C) = \sum_{y \in Y} \text{min}_{i=1..k} ||y - c_i||^2 \tag{2}$$

Целью кластеризации k-средних является выбор набора C из k центров для минимизации $\phi_X(C)$.

Основным недостатком инициализации k-means++ с точки зрения масштабируемости является присущий ей последовательный характер: выбор следующего центра зависит от текущего набора центров. Также необходимо знать изначальное количество кластеров.

1.4.2 Нейронные сети

Нейронные сети позволяют быстро, а также эффективно решать задачи кластеризации. Основное преимущество нейронных сетей заключается в том, что они хорошо приспособлены для параллельных вычислений и обучаемые.

Качество нейронной сети зависит от данных, на которых она обучается, и от того, насколько верно подобрали ее структуру. Предобработка данных [12] – это преобразование статистического набора данных.

На входы нейронной сети подаются значения признаков выбранного объекта. Нейросеть обрабатывает эти сигналы, после чего в выходном слое определяется нейрон-победитель. Нейрон-победитель выходного слоя определяет класс объекта, признаки которого были поданы на входы нейросети.

Такой подход к кластеризации особенно необходим при работе с большими объемами данных, требующими больших затрат вычислительной мощности и машинного времени.

Преимущества данного метода:

1) устойчивость к шумам входных данных;
2) адаптация к изменениям;
3) отказоустойчивость;
4) быстродействие.

Недостатки:

1) неточность ответа
2) принятие решений в несколько этапов;
3) вычислительные задачи.

1.4.3 Fuzzy C-means

Существует множество методов нечеткой кластеризации. Среди них широко используется алгоритм нечетких С-средних (FCM). Он основан на концепции нечеткого С-разбиения. Данный алгоритм и его производные очень успешно использовались во многих приложениях, таких как распознавание образов, классификация, интеллектуальный анализ данных и сегментация изображений.

Обычно алгоритм C-means состоит из нескольких этапов выполнения. На первом шаге алгоритм случайным образом выбирает С начальных центров кластера из исходного набора данных. Затем, на более поздних этапах, после некоторых итераций алгоритма, конечный результат сходится к фактическому центру кластера. Поэтому выбор хорошего набора начальных центров кластера очень важен для алгоритма FCM. Однако трудно случайным образом выбрать хороший набор начальных кластерных центров. Если выбран хороший набор начальных центров кластера, алгоритму может потребоваться меньше итераций, чтобы найти фактические центры кластера.

Нечеткий алгоритм c-means минимизирует величину

$$\sum_{i=1}^{|X|}\sum_{j=1}^{C} u_{i,j}^{m}||x_i - c_j||^2, 1 \leq m \leq \infty, \tag{3}$$

ге $m \in R$, $u_{i,j}$ - коэффициент принадлежности вектора x_i к кластеру c_j, x_i - i-ый компонент $|X|$-мерного вектора X, C - количество кластеров, c_j - центр j-го

кластера, а $||*||$ - норма, которая определяет расстояние от вектора до центра кластера.

Преимуществом данного алгоритма является то, что он является нечетким и каждый из объектов принадлежит всем кластерам с разной степенью принадлежности.

Недостатками является то, что из-за того, что данный алгоритм является нечетким, он требует больших вычислительных затрат. Также необходимо заранее знать количество кластеров. Алгоритм очень чувствителен к выбору начальных центров кластеров.

1.4.4 Гауссовы смеси

Существует категория методов кластеризации, которые определяют кластеры как наблюдения, имеющие, скорее всего, одинаковое распределение [13]. В этом последнем случае предполагается, что каждая субпопуляция распределена по параметрической плотности, подобной гауссовой, и, таким образом, неизвестная плотность данных представляет собой смесь этих распределений.

На практике каждый кластер представлен параметрическим распределением, подобным гауссову, и весь набор данных моделируется смесью этих распределений [14]. Преимущество кластеризации на основе моделей заключается в обеспечении строгой структуры для оценки количества кластеров и роли каждой переменной в процессе кластеризации

Чем больше информации у нас есть о каждом человеке, тем лучше ожидается, что метод кластеризации будет работать. Однако структура, представляющая интерес, часто может содержаться в подмножестве доступных переменных, и многие переменные могут быть бесполезными или даже вредными для обнаружения разумной структуры кластеризации. Таким образом, важно выбрать соответствующие переменные с точки зрения кластерного анализа.

Распределение Гаусса, также называемое нормальным распределением,

представляет собой непрерывное распределение вероятностей:

$$N(X|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}\sqrt{|\Sigma|}} \exp -\frac{(X-\mu)^T\Sigma^{-1}(X-\mu)}{2}, \quad (4)$$

где μ - D-мерный средний вектор, Σ - D x D ковариационная матрица, которая описывает форму Гаусса и $|\Sigma|$ обозначает определитель Σ.

Модель Гауссовых смесей представляется в виде линейной комбинации базового распределения веротностей по Гауссу и выражается как

$$p(X) = \sum_{k=1}^{K} \pi_k N(X|\mu_k, \Sigma_K) \quad (5)$$

1.4.5 Вывод

Возможность обучения является одним из главных преимуществ нейронных сетей перед остальными методами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искажённых данных.

Также на вход нейронной сети можно передавать не предобработанные (сырые) данные. У других рассмотренных методов основным недостатком является неустойчивость к шумам.

1.5 Классификация нейронных сетей для решения задачи кластеризации пользователей социальных сетей

1.5.1 Сети адаптивного резонанса

Теория адаптивного резонанса (APT) имеет биологическую мотивацию и является крупным достижением в парадигме конкурентного обучения [15]. Теория приводит к серии неконтролируемых сетевых моделей в реальном времени для кластеризации, распознавания образов и ассоциативной памяти.

Модели способны к стабильному распознаванию категорий в ответ на произвольные входные последовательности с быстрым или медленным обу-

чением. Модели ART характеризуются системами дифференциальных уравнений, которые формулируют устойчивые самоорганизующиеся методы обучения.

На этапе обучения сохраненный прототип категории адаптируется, когда шаблон ввода достаточно похож на прототип. Когда обнаруживается новизна, ART адаптивно и автономно создает новую категорию с исходным шаблоном в качестве прототипа.

Основным модулем обработки любой сети ART является конкурентоспособная обучающая сеть. Нейроны m входного слоя F_1 регистрируют значения входного шаблона $I = (i_1, i_2, ..., i_m)$. Каждый нейрон выходного слоя F_2 получает восходящую сетевую активность t_j, построенную из всех выходов F_1. Векторные элементы $T = (t_1, ..., t_n)$ можно рассматривать как результаты сравнения между входным шаблоном I и прототипами $W_1 = (w_{11}, ..., w_{1m}), ..., W_n = ((w_{n1}, ..., w_{nm})$. Эти прототипы хранятся в синаптических весах соединений между F_1 и F_2-нейронами. Единственный F_2-нейрон J, получающий самую высокую чистую активность t_J, устанавливает свой выходной сигнал равным единице, в то время как все остальные выходные нейроны остаются равными нулю

$$u_i = \begin{cases} 1 & \text{если } t_j > \max(t_k : k \neq j) \\ 0 & \text{иначе.} \end{cases} \tag{6}$$

Одним из возможных способов вычисления чистой активности и с помощью этого измерения сходства между и является взвешенная сумма

$$t_j = \sum_{i=1}^{m} w_{ij} i_i \tag{7}$$

Часто используются вариации этого показателя, поскольку значение оказывает большое влияние на результирующие кластеры. После того, как F_2 победитель J был найден, соответствующий прототип $W_J = (w_{iJ}, ..., w_{mJ})$ адаптируется к входному шаблону I. Одним из подходящих методов адаптации является небольшое смещение в сторону входного шаблона.

$$W_J^{\text{new}} = \eta I + (1 - \eta) W_J^{\text{old}} \tag{8}$$

Недостатками данной сети является то, что она имеет большое количество синаптических связей в сети. При этом многие из обучающих весов после обучения оказываются нулевыми. Также результат часто зависит от порядка обучающей выборки.

1.5.2 Сети Кохонена

Сети (слои) Кохонена относятся к самоорганизующимся нейронным сетям [16]. Самоорганизующаяся сеть позволяет выявлять кластеры (группы) входных векторов, обладающих некоторыми общими свойствами.

Кластеризация позволяет сгруппировать сходные данные, что облегчает решение ряда задач Data Mining:

1) изучение данных, облегчение анализа;
2) прогнозирование;
3) обнаружение аномалий.

С помощью сетей Кохонена производится кластеризация объектов, описываемых количественными характеристиками.

Сеть (слой) Кохонена (рисунок 2) — это однослойная сеть, построенная из нейронов типа WTA (Winner Takes All — победитель получает все).

Для обучения сети применяются механизмы конкуренции. Перед процессом обучения производится инициализация сети, то есть первоначальное задание векторов весов. В простейшем случае задаются случайные значения весов. Процесс обучения сети Кохонена состоит из циклического повторения ряда шагов:

1) подача исходных данных на входы;
2) нахождение выхода каждого нейрона;
3) определение «выигравшего» нейрона;
4) корректировка весов «выигравшего» нейрона по правилу Кохонена;

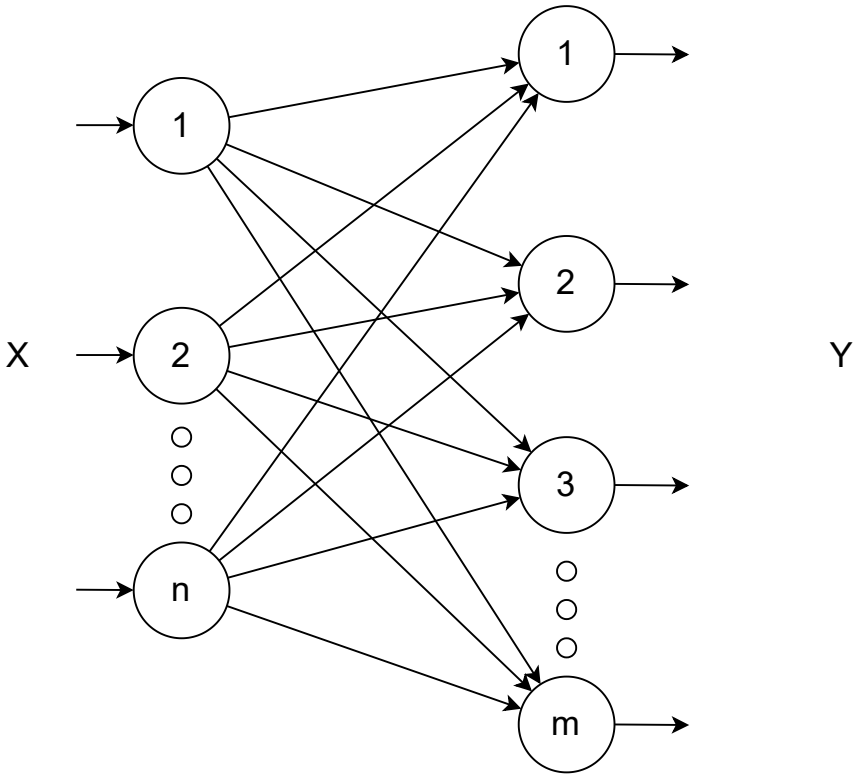

Рисунок 2 – Структура сети Кохонена[17]

5) переход на шаг 1, если обучение не завершено.

Алгоритм учитывает евклидово расстояние между двумя n-мерными векторами, которое измеряется сходством между входными векторами [18]. Расстояние входного вектора от каждого нейрона i, D_i задается формулой

$$D_i = ||W_{ij} - X|| = \sqrt{\sum_{j=1}^{n}(x_j - w_{ij})^2} \tag{9}$$

где $X = (x_1, ..., x_n)^{\mathrm{T}}$ обозначает входной вектор $w_{ij} = (w_{11}, ..., w_{im})^{\mathrm{T}}$.

Победителем объявляется Кохонен с минимальной дистанцией. Другими словами, вектор веса победителя находится ближе всего к входному вектору.

$$D_w = \min\{D_i\}, i \in \{1, 2, ..., m\} \tag{10}$$

Во время обучения победитель настраивает свои веса так, чтобы они были ближе к значениям данных, а соседи победителя также настраивают свои веса так, чтобы они были ближе к тому же вектору входных данных в соответствии со следующим соотношением

$$W_i = W_{ij} + \alpha(W_i - X), i = \{1, 2, ..., m\} \quad (11)$$

Таким образом, части сети конкурируют за выбор. Только веса победителя будут адаптированы. Настройка соседнего нейрона играет важную роль в сохранении порядка входных данных. Таким образом, выигравший нейрон находится ближе всего к входному значению. После обучения весовые векторы самоорганизуются и представляют собой прототипы классов входного вектора.

1.5.3 Перцептрон без учителя

В основе перцептрона лежит математическая модель восприятия информации мозгом. Разные исследователи по-разному его определяют. В самом общем своем виде (как его описывал Розенблатт) он представляет систему из элементов трех разных типов: сенсоров, ассоциативных элементов и реагирующих элементов.

Перцептрон стал одной из первых моделей нейросетей. На рисунке 3 представлена схема перцептрона.

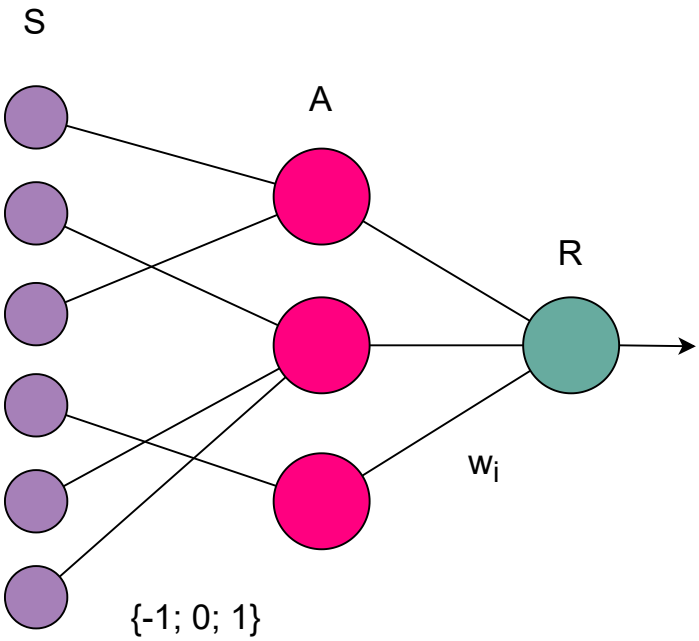

Рисунок 3 – Схема перцептрона

Кроме классического метода обучения перцептрона, Розенблат также ввел понятие об обучении без учителя, предложив следующий способ обучения: аль-

фа-система подкрепления – это система подкрепления, при которой веса всех активных связей, ведущих к элементу, изменяются на одинаковую величину r, а веса неактивных связей за это время не меняются.

Позже, с разработкой понятия многослойного перцептрона, альфа-система была модифицирована, и ее стали называть дельта-правилом. Модификацию было проведено с целью сделать функцию обучения дифференцируемой (например, сигмоидною), что в свою очередь требуется для применения метода градиентного спуска, благодаря которому возможно обучение более одного слоя.

В модели перцептрона используется один нейрон с линейной взвешенной сетевой и пороговой функциями активации. Входным сигналом для этого нейрона $x = (x_1, ..., x_n)$ является вектор признаков в n-мерном пространстве признаков. Функция $f(x)$- это взвешенная сумма входных данных:

$$f(x) = w_0 + \sum_{i=1}^{n} w_i x_i \tag{12}$$

Обучение - это процесс, посредством которого свободные параметры нейронной сети адаптируются посредством непрерывного процесса стимуляции со стороны среды, в которую встроена сеть [19]. Тип обучения определяется способом, которым происходят изменения параметров.

Алгоритм обучения перцептрона может быть реализован на электронном устройстве, и сеть становится в определенном смысле самоподстраивающейся. По этой причине процедуру подстройки весов обычно называют «обучением» и говорят, что сеть «обучается».

1.5.4 Вывод

В качестве нейронной сети была выбрана сеть Кохонена, так как заранее известно необходимое число кластеров. Немало важным фактором является устойчивость к шумам. Из-за того, что в выборке будут часто встречаться шумы, так как пользователи не всегда указывают полную информацию о себе и часто опускают некоторые данные, сеть Кохонена является наиболее подходящей для задачи персонализации пользователей социальных сетей для целевой

рекламы.

1.6 Выводы

В данном разделе были проанализированы существующие программные решения, оказавшиеся в закрытом доступе, на основании чего было принято решение о разработке собственного метода персонализации пользователей социальных сетей.

Сравнительный анализ социальных сетей для решения задачи получения информации о пользователях выявил целесообразность использования API социальной сети ВКонтакте, так как оно является открытым и сама сеть не заблокирована на территории Российской Федерации.

Для решения задачи кластеризации была выбрана нейронная сеть Кохонена из-за ее устойчивости к шумам, высоком быстродействии, а также возможности обучения без начального указания общего числа кластеров.

2 Конструкторский раздел

В этой части описаны методы и алгоритмы, использованные в разработке, и описан реализованный метод. Также в этом разделе подробно изложена архитектура разработанной программы, представлена архитектура нейронной сети и приведены способы обучения этих сетей.

2.1 Функциональная модель

На рисунке 4 изображена функциональная модель персонализации пользователей социальной сети с использованием нейронных сетей в нотации IDEF0.

На вход подается идентификатор пользователя социальной сети ВКонтакте. С помощью открытого API обрабатывается необходимая, а также открытая информация о пользователе. После этого происходит предобработка полученных данных и персонализация пользователя с помощью обученной нейронной сети.

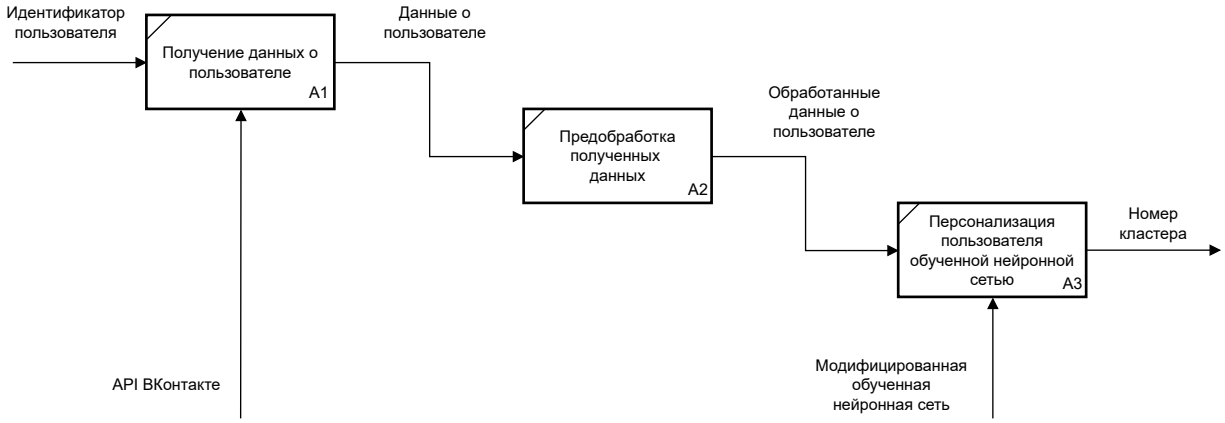

Рисунок 4 – IDEF0 диаграмма второго уровня

Результат работы программы - полученная тематика целевой рекламы для выбранного пользователя.

2.2 Получение информации о пользователях социальных сетей

API ВКонтакте — это интерфейс, который позволяет получать информацию из базы данных ВКонтакте с помощью запросов к специальному серверу. Синтаксис запросов и тип возвращаемых ими данных строго определены на

стороне самого сервиса.

Методы API - команды, которые позволяют работать с определенными операциями базы данных ВКонтакте.

Для сбора информации о пользователях понадобятся следующие методы:

1) users.search – возвращает список пользователей в соответствии с заданным количеством;
2) users.get – возвращает расширенную информацию о пользователях;
3) groups.get – возвращает список сообществ указанного пользователя.

Для того, чтобы можно было выяснять, какая тематика интересует определенного пользователя – просматривается список групп, на которые подписан пользователь и выбирается тема, которая наиболее чаще повторяющаяся среди групп пользователя.

На рисунке 5 представлен алгоритм получения информации о различных пользователях социальных сетей.

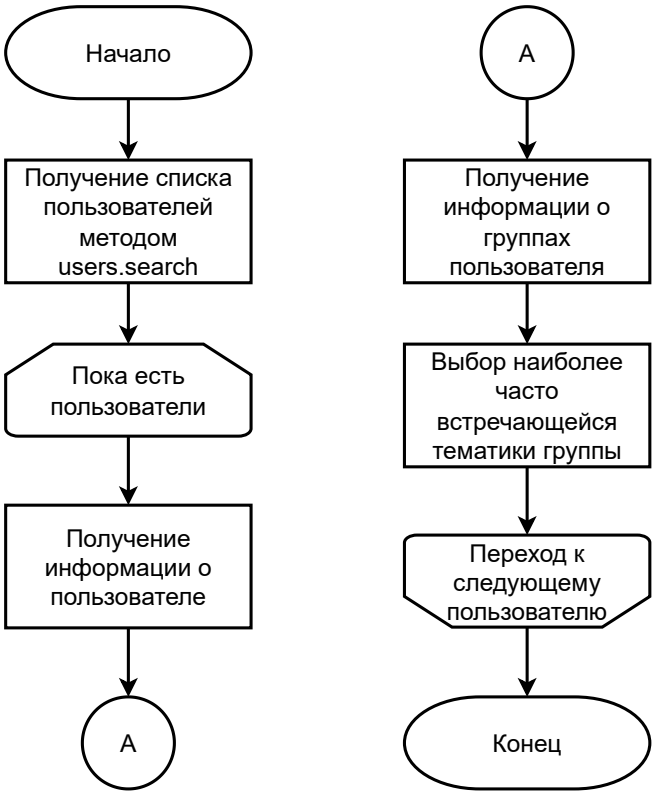

Рисунок 5 – Алгоритм получения информации о пользователях

В приложении А представлен пример результата запроса users.get по идентификатору пользователя «mrsklif».

В нем представлено следующее: bdate – дата рождения, career – информацию о карьере, city – город проживания, country – страну проживания, followers_count – количество подписчиков, education – информацию об образовании, has_mobile – есть ли телефон, has_photo – есть ли фотографии, counters – различные счетчики (количество друзей, групп, страниц), home_town – город рождения, military – информации о прохождении службы, personal – различную информацию о жизненной позиции, relation – информация об отношениях, sex – пол человека, timezone – временная зона.

2.3 Формирование обучающей выборки

Если хотя бы один из векторов подвергается нормализации, то процесс самоорганизации приводит к связному разделению пространства данных.

Нормализация векторов достигается увеличением размерности на одну координату ($R^N \rightarrow R^{N+1}$) с таким выбором значения $(N+1)$-го компонента вектора, чтобы

$$\sum_{i=1}^{N+1} x_i^2 = 1 \tag{13}$$

Алгоритм нормализации векторов обучающей выборки представлен на рисунке 6.

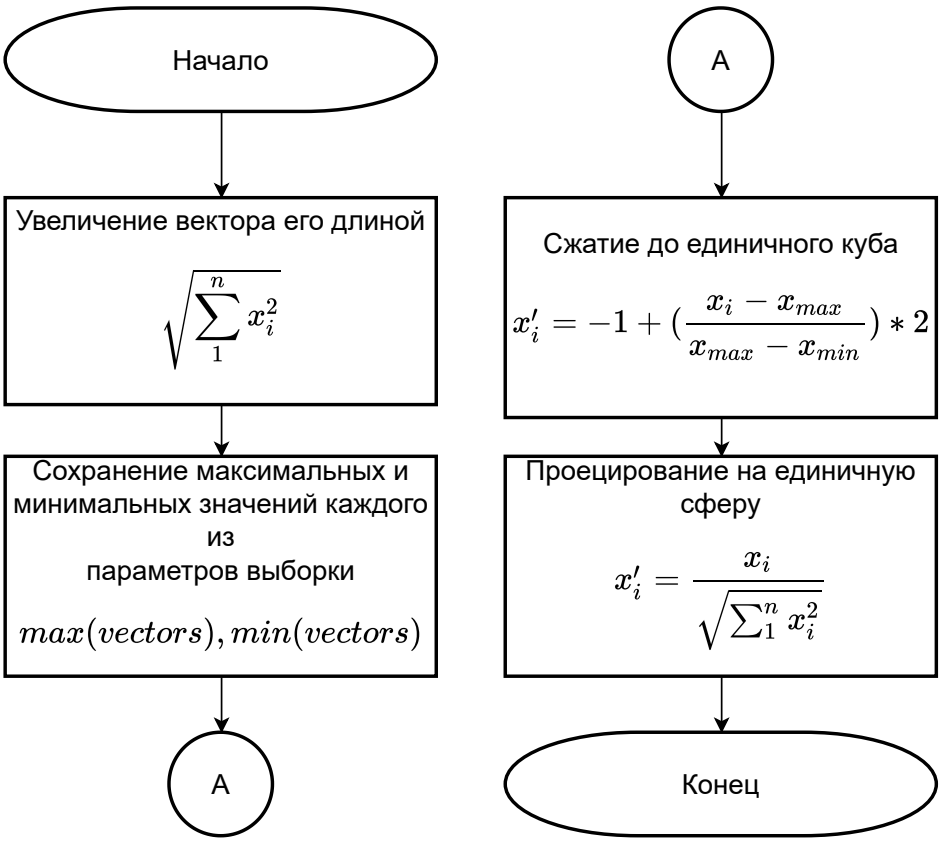

Рисунок 6 – Схема алгоритма формирования обучающей выборки

2.4 Архитектура нейронной сети Кохонена

На рисунке 7 представлена архитектура нейронной сети Кохонена.

На данном рисунке введены следующие параметры:

1) n – размерность входного вектора;
2) m – количество выходов.

Данная нейронная сеть является однослойной и состоит из нейронного слоя Кохонена.

Результатом работы проектируемой нейронной сети является индекс нейрона победителя, то есть номер интересной пользователю тематики.

Выходными данными нейронной сети в процессе обучения является вектор, а также номер кластера (номер тематики, которая интересна пользователю), к которому предположительно относится пользователь.

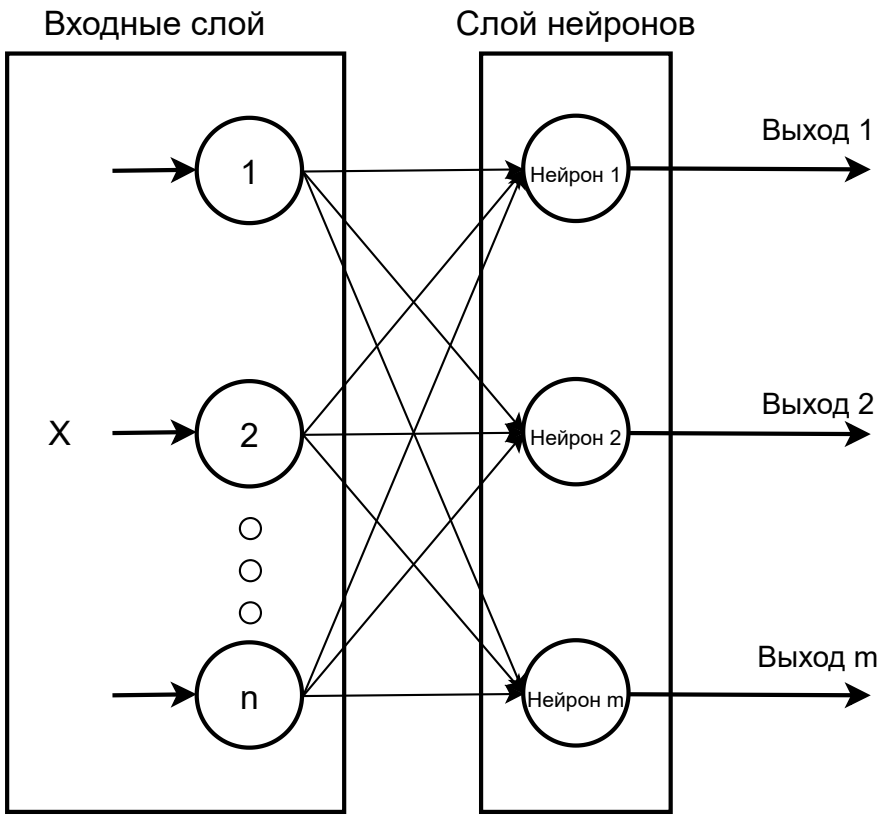

Рисунок 7 – Архитектура нейронной сети Кохонена

2.5 Обучение нейронной сети

Перед обучением нейронной сети необходимо проинициализировать веса нейронов, далее при каждой итерации получать индекс «победившего» нейрона по входному примеру и корректировать веса этого нейрона.

2.5.1 Инициализация весов

Метод выпуклой комбинации позволяет правильно распределить плотность векторов весов в соответствии с плотностью входных векторов в пространстве X.

Изначально необходимо присвоить всем весам одно и то же начальное значение:

$$w_j^k = \frac{1}{\sqrt{m}} \tag{14}$$

В данном выражении m – количество параметров входных векторов. Вектора весов получают длину, равную единице, как требует нормировка.

Далее необходимо проводить обучение с векторами:

$$x'_j = x_j * \alpha + \frac{1-\alpha}{\sqrt{m}} \tag{15}$$

В данном выражении m – количество параметров входных векторов, α – коэффициент сжатия.

Метод выпуклой комбинации позволяет получить правильное распределение плотности ядер. В нейронной сети не остается «ненужных» необученных нейронов. Когда вектор нейрона находится далеко от обучающих векторов, то он не будет «победителем», и его веса не будут корректироваться при обучении.

2.5.2 Алгоритм обучения нейронной сети

Основная задача обучения нейронной сети – научить сеть активировать один и тот же нейрон для схожих векторов на входе.

Изначально необходимо проинициализировать веса нейронов. Обычно такие веса выбираются малыми случайными числами, но для слоя Кохонена такой выбор имеет недостатки. Если веса будут проинициализированы сучайными значениями с равномерным распределением нейронов, то в областях пространства, где мало входных векторов, нейроны будут использоваться редко.

Для устранения данной проблемы необходимо использовать метод выпуклой комбинации.

Для обучения сети необходимо настраивать веса итеративным алгоритмом, при котором корекции весов проводятся после предъявления каждого входного вектора, а не после предъявления всех.

Алгоритм:

1) присваиваем начальные значения весовым коэффициентам;
2) подаем на вход один из векторов обучающей выборки;
3) рассчитываем выход слоя Кохонена и определяем номер выигравшего нейрона, выход которого максимален;
4) корректируем веса только выигравшего нейрона.

Веса необходимо корректировать так, что вектор весов приближается к

текущему входному вектору.

Скорость обучения управляет быстротой приближения ядра вектора весов ко входному вектору.

Алгоритм необходимо выполнять, пока веса не перестанут меняться.

Также присутствует необходимость в нормализации входных векторов. Она обусловлена тем, что значения признаков могут находиться в большом диапазоне и отличаться на несколько порядков. При нормализации все значения признаков будут приведены к одинаковой области их изменения, что обеспечит корректную работу алгоритма.

На рисунке 8 представлена схема алгоритма обучения нейронной сети.

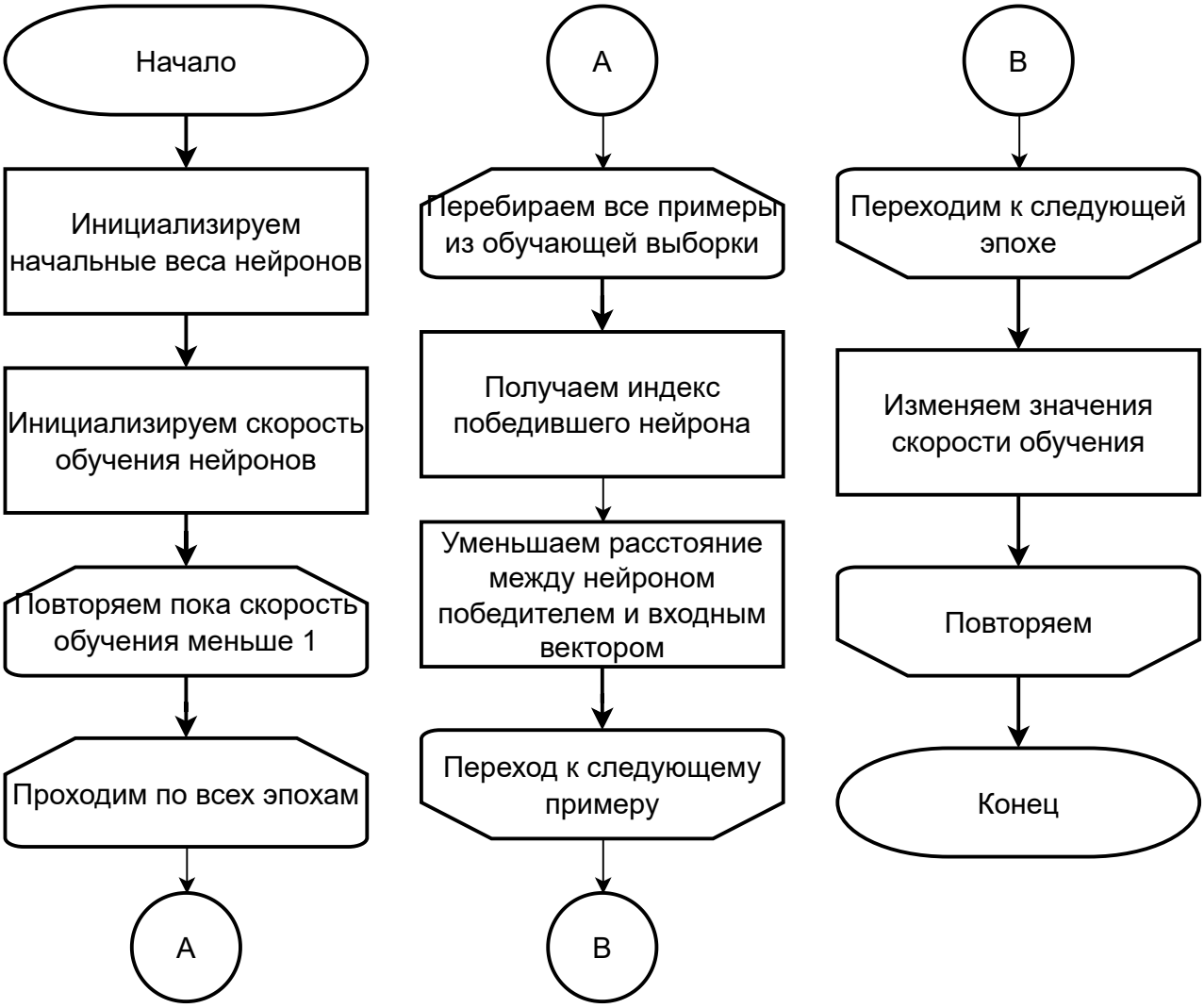

Рисунок 8 – Схема алгоритма обучения

2.5.3 Метод определения нейрона «победителя»

Метод определения нейрона «победителя» основан на вычислении евклидова расстояния.

На вход подается вектор из обучающей выборки. Для корректного обучения нейронной сети первым вектором берется произвольный пример из обучающей выборки, а после корректировки весов обучающей вектор берется максимально удаленный от предыдущего (вычисляется расстояние Евклида).

Расстояние Евклида — это геометрическое расстояние в многомерном пространстве:

$$D_j = \sqrt{\sum_{i=1}^{n}(y_i - w_{i,j})^2}, \tag{16}$$

где y_i - входное воздействие, $w_{i,j}$ - весовой коэффициент, на которое попадает данное воздействие.

После применения данной формулы получается массив расстояний между входным воздействием и весовыми коэффициентами, то есть расстояние между нейронами и входным воздействием. «Победителем» является тот, чье расстояние наименьшее.

При первых итерациях обучения сети выбираются несколько нейронов, у которых расстояние наименьшее.

После определения «победившего» нейрона осуществляется корректировка его веса:

$$w'_{i,j} = w_{i,j} + \eta * [y_i - w_{i,j}] \tag{17}$$

На рисунке 9 представлена схема алгоритма поиска нейрона победителя.

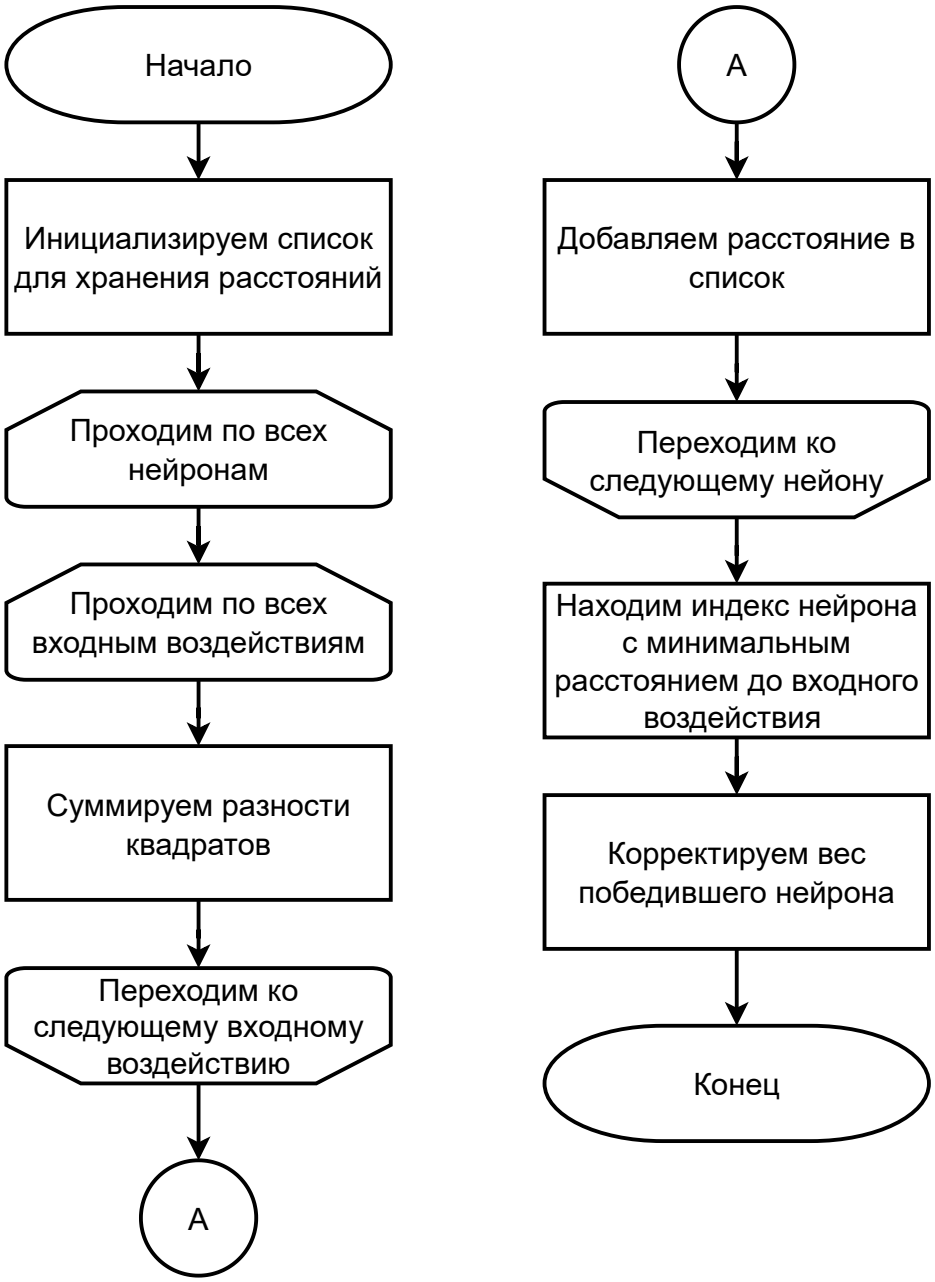

Рисунок 9 – Схема алгоритма поиска нейрона «победителя»

2.6 Модификация алгоритма

Следует принять во внимание, что может возникнуть две ситуации:

1) некоторые данные, находящиеся далеко от центра сферы и относящиеся к разным кластерам при проецировании на сферу попадают в один кластер;
2) при проецировании кластера на сферу получилось так, что кластер попал в центр единичной сферы.

В таких ситуациях необходимо разбить получившийся кластер на более мелкие. Такой подход называется нейросетевой каскадной кластеризацией данных.

Предлагается следующая модификация: изначально строится «грубая» сеть Кохонена и для получившихся «смешанных» кластеров (такие, у которых наибольшее количество данных из обучающей выборки) строятся дополнительные сети Кохонена.

Таким образом, благодаря такой модификации получается каскад. На рисунке 10 представлена каскадная нейронная сеть Кохонена.

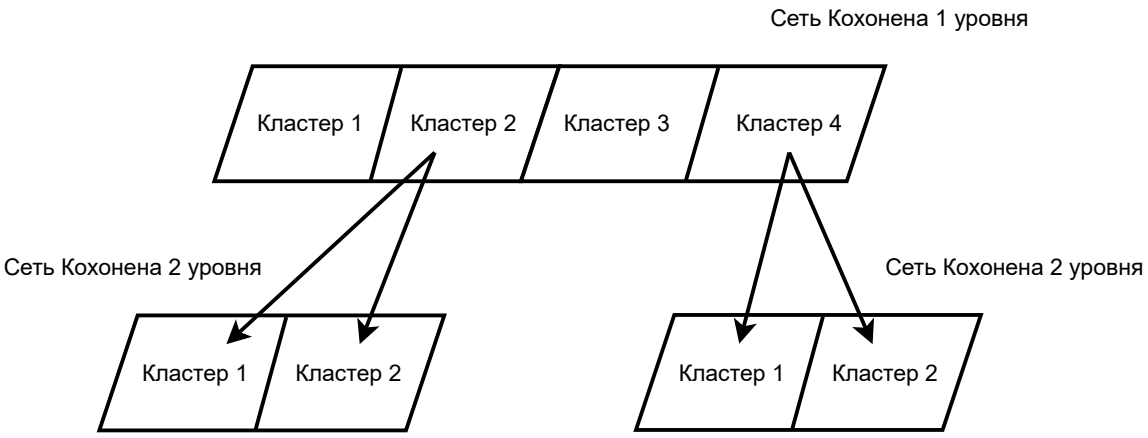

Рисунок 10 – Схема алгоритма поиска нейрона «победителя»

2.7 Выводы

Таким образом, в данном разделе были разработаны архитектура программного продукта и нейронной сети Кохонена, алгоритмы получения информации о пользователях социальных сетей и обучения нейронной сети. Также

была предложена модификация нейронной сети Кохонена для более точной кластеризации пользователей.

3 Технологический раздел

В данном разделе будут указаны средства разработки и требования к вычислительной системе, расписана архитектура программного обеспечения.

Будут реализованы алгоритмы для сбора информации о пользователях социальной сети ВКонтакте, для обучения каскадной нейронной сети Кохонена.

Будет продемонстрирована работа программного обеспечения и описан функционал интерфейса.

3.1 Средства разработки

Для получения информации о пользователях социальной сети ВКонтакте, нормализации полученных данных, разработки и обучения каскадной нейронной сети Кохонена исользовался следующий технологический стек:

- Язык программирования Python.

  Для написания программ на данном языке программирования требуется писать меньший объекм кода. Это ускоряет процесс разработки.

  Для Python доступно огромное количество дополнительных библиотек.
- Среда разработки PyCharm.

  Данная среда разработки является бесплатной. Также она позволяет быстро производить рефакторинг кода, а также использовать удобный графический отладчик.
- Библиотеки json, urllib.

  Данные библиотеки позволяют взаимодействовать с открытым API ВКонтакте, а также быстро обрабатывать полученную информацию о пользователях.
- Прочие библиотеки numpy – для ускорения обучения нейронной сети и работы с данными, matplotlib и pylab для построения графиков, PyQt5 - для графического интерфейса.

3.2 Требование к вычислительной системе

Программа обрабатывает информацию о пользователях социальной сети в большом объеме, поэтому занимаемая программой память может быть от 1 до 4 ГБ.

Специальных требований к центральному процессору не предъявляется, но его частота напрямую влияет на производительность.

Программа была разработана, обучена и протестирована на компьютере со следующими характеристиками:

1) процессор Intel Core i5-10210U с базовой частотой 1.6 МГц с поддержкой технологии Turbo Boost;
2) объем оперативной памяти: 32Гб;
3) операционная система: Ubuntu 18.

3.3 Архитектура программного обеспечения

Программный продукт состоит из 8 модулей:

1) Модуль формирования запросов к API ВКонтакте.

   В данном модуле составляются запросы с необходимыми параметрами, которые должны возвращать информацию о пользователях.

2) Модуль получения информации о пользователях.

   Данный модуль отвечает за отправку запросов и получение ответов от открытого API ВКонтакте.

3) Модуль нормализации данных.

   Необработанные данные о пользователях поступают на вход данного модуля.

4) Модуль обучения нейронной сети.

   На вход данного модуля поступают нормализованные данные о пользователях. Алгоритм работы данного модуля описан на рисунке 8.

5) Модуль построения каскадной обученной нейронной сети.

   При построении каскадной нейронной сети сначала указывается базовая

(«грубая») нейронная сеть и проверяется на возможность разбиения на более мелкие кластеры.

6) Модуль для взаимодействия с каскадной нейронной сетью.
7) Модуль для взаимодействия с пользователем.

   Данный модуль отвечает за графический интерфейс, который дает возможность обучать, загружать каскадную нейронную сеть, персонализировать пользователя и проводить эксперименты.

Архитектура программного обеспечения указана на рисунке 11.

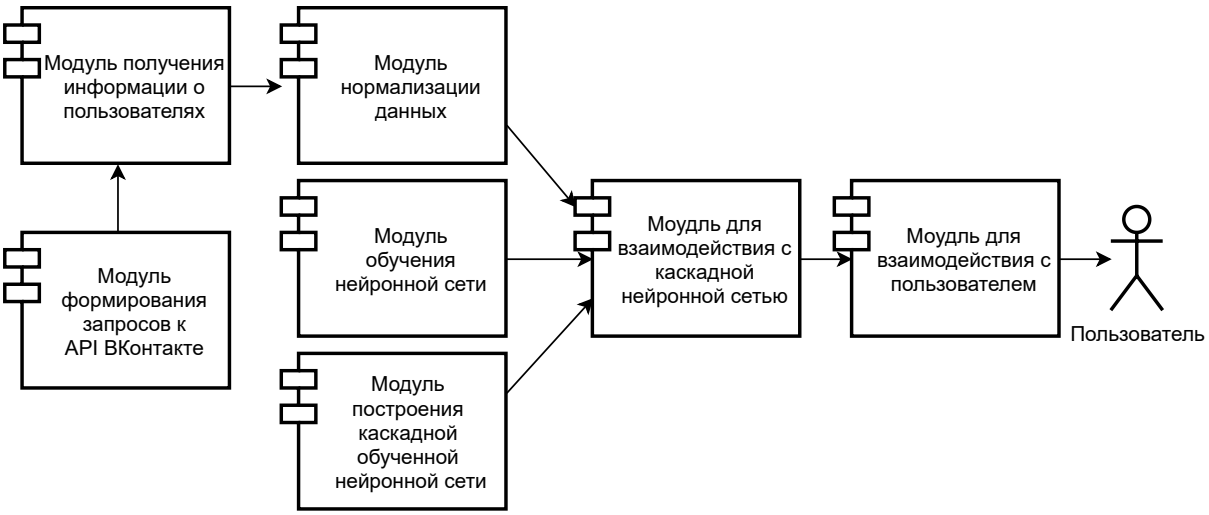

Рисунок 11 – Архитектура ПО

3.4 Сбор информации о пользователях социальной сети ВКонтакте

API вконтакте позволяет получать необходимую информацию о пользователях с помощью запроса user.search. Пример такого запроса представлен в приложении А. Проблема заключается в том, что без заданных параметров возможно получиться максимум 1000 человек, поэтому было принято решение получать информацию с заданным городом проживания человека, что позволяет получать различные данные в большом объеме.

Пример запроса для получения информации о пользователе представлен в листинге 1.

Листинг 1: Пример запроса

```
https://api.vk.com/method/users.search?access_token=TOKEN&v=5.131&
count=999&city=1&fields=activities,bdate,career,city,country,education,
followers_count,counters,has_mobile,has_photo,home_town,military,online,
personal,relation,sex,timezone
```

Результат получение информации о пользователях представлен в листинге 2.

Листинг 2: Пример полученной информации

```
1;1;2;0;21;1;1;1;1;105;0;250;1033;0;0;0;2;3;2;2;2
2;1;2;0;1;1;1;1;1;49;0;250;0;0;0;0;0;0;0;0;6
3;1;1;0;23;1;1;1;1;124;0;0;0;0;0;3;6;3;2;0;6
4;1;2;0;13;0;1;1;1;201880;0;0;0;0;0;0;0;0;0;0;19
5;1;2;1;0;1;1;1;1;7;1;250;1033;2022;0;0;0;0;4;4;3
6;1;2;0;0;0;1;0;1;4;0;0;0;0;0;0;0;0;0;0;1
7;1;2;1;31;1;1;1;1;86;0;250;1033;2014;0;5;2;6;1;1;19
8;1;1;0;16;1;1;1;1;70627;1;2;27;0;0;0;0;0;0;0;2
9;1;2;1;23;61;1;1;1;271;0;0;0;0;0;0;0;0;0;0;0
10;1;2;0;21;1;1;1;1;89;0;250;1033;2022;0;0;0;0;0;0;6
```

В результате была собрана информация о 90361 человеке.

3.5 Обучение каскадной нейронной сети Кохонена

Для обучения каскадной нейронной сети Кохонена изначально необходимо выделить базовые кластеры. Сначала считываются и нормализуются данные о пользователях социальной сети. Далее инициализируются начальные параметры нейронной сети:

1) устанавливаются веса нейронов: $w_i = 1/\sqrt{m}$, где m – количество параметров одного примера обучающей выборки;
2) устанавливается начальное значение скорости обучения: $\eta = 1$, изменяющееся по формуле: $\eta = \eta 0/sqrt(t+1)$;
3) устанавливается количество эпох;
4) устанавливается коэффициент сжатия: $\alpha = 0.01$, изменяющееся по формуле: $\alpha = \alpha 0 * \sqrt{t+1}$;
5) устанавливается количество нейронов, которое нужно корректировать за 1 раз при обучении: $cNeurons = 4$, которое меняется по формуле: cNeurons = cNeurons0$/\sqrt{t+1}$.

В приведенных формулах t – параметр, который увеличивается пропорционально прохождению всех эпох на заранее заданную фиксированную величину.

Первый входной вектор каждой эпохи выбирается произвольно из обучающей выборки, а для получению следующего вектора производятся следующие действия:

1) выбираются произвольно несколько векторов из обучающей выборки;
2) рассчитываются расстояние Евклида до изначального входного вектора;
3) выбирается вектор с максимальным расстоянием до изначального входного вектора.

При корректировке весов выбираются несколько нейронов с минимальным расстоянием Евклида до входного вектора.

Далее вычисляется количество примеров из обучающей выборки для каждого кластера, удаляются мертвые нейроны и вычисляется дисперсия каждого кластера. После этого сохраняются полученные веса кластеров.

После обучения базовой нейронной сети Кохонена выделяются кластеры с наибольшим количеством вошедших в него примеров обучающей выборки и обучается новая сеть с обучающей выборкой, состоящей из примеров, попавших в соответствующий кластер.

В листинге 3 представлены веса одного из нейронов.

Листинг 3: Пример весов

```
0.10281263151999545;-0.23424505742066823;-0.11347980294860255;
-0.2237547563553921;-0.2320872759696989;-0.23424505742066823;
0.23876385749776988;-0.23213043868682237;-0.23424505742066823;
-0.23255659058157185;-0.22950123463618785;-0.23424505742066823;
-0.22354657004415016;-0.2342318201060587;-0.20888743981696106;
-0.22151503135904085;-0.2257089196516377;-0.23026024303627338;
-0.22124676059242473
```

3.6 Формат входных и выходных данных

3.6.1 При обучении нейронной сети

На вход подаются данные о пользователях социальной сети в формате csv. В файле должны находиться параметры типа int. Информация об 1 пользователе находится на одной строке.

В качестве выходных данных является дирректория, в которой хранятся веса базовой нейронной сети, а также поддиректория, в которой хранятся веса сетей, которые необходимо было дополнительно обучить для большего разделения. Названия файлов в поддиректории соответствуют индексам кластеров, которые необходимо было дополнительно «разбить».

3.6.2 При персонализации пользователя

На вход подается идентификатор пользователя, который можно найти в ссылке на профиль пользователя социальной сети Вконтакте. Например, идентификатор «mrsklif», полученный из ссылки: «https://vk.com/mrsklif».

В качестве выходных данных - идентификатор кластера, в который попал пользователь.

3.7 Демонстрация работы

При запуске программного обеспечения появляется графическое окно. В

1) Обучение каскадной нейронной сети.

   В данной группе можно задать начальную сколько обучения η, начальное количество нейронов для корректировки cNeurons, начальное значение коэффициента сжатия α и начальное количество нейронов сети n.

2) Загрузка обученной нейронной сети.

   В данной группе можно указать директории базовой сети и сетей второго уровня. В этих директории сохраняются файлы после обучения. Также после загрузки выводятся полученное число нейронов базовой сети и общее число нейронов каскадной сети.

3) Персонализация пользователя.

   Перед персонализацией пользователя необходимо загрузить каскадную нейронную сеть. В данной группе доступно поле ввода для идентификатора, а также выводятся получившиеся кластеры базовой сети и кластера второго уровня.

4) Исследования.

   В данной группе можно вывести диаграмму Эндрюса, зависимость разбиения кластеров второго уровня от их среднеквадратичного отклонения, попавших в кластер базовой сети и исследовать срднеквадратичное отклонение всей выборки, базовой и каскадной сети.

Интерфейс программного обеспечения продемонстрирован на рисунке 12.

Также были реализованы алгоритмы сбора информации о пользователях социальной сети ВКонтакте и обучения каскадной нейронной сети Кохонена.

Была продемонстрирована работа программного продукта.

3.8 Выводы

В данном разделе были указаны средства разработки и требования к вычислительной системе, расписана архитектура программного обеспечения, реализованы алгоритмы для сбора информации о пользователях социальной сети ВКонтакте, для обучения каскадной нейронной сети Кохонена, продемонстрирована работа программного обеспечения и описан функционал интерфейса.

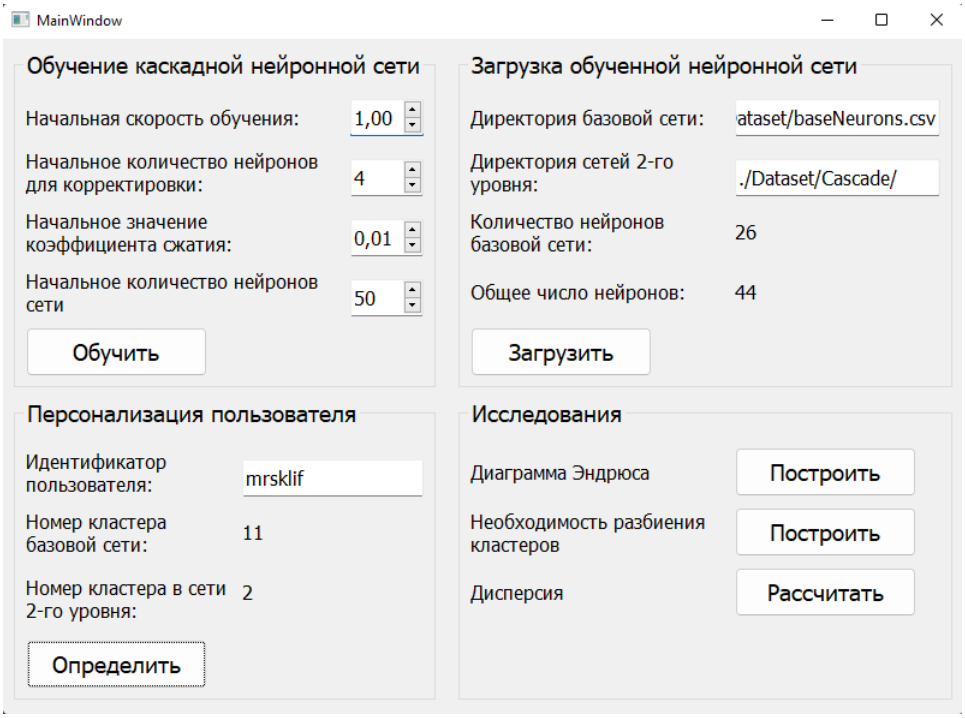

Рисунок 12 – Интерфейс программного обеспечения

В качестве языка программирования разработанного программного обеспечения был выбрал язык Python, средой разработки – PyCharm. Был определен формат входных и выходных данных.

Демонстрация работы показала работоспособность реализованного программного обеспечения.

4 Исследовательский раздел

В данном разделе будут собраны данные о пользователях для исследования, построена диаграмма Эндрюса для исследования корректного разбиения на кластеры, исследована зависимость количества выделенных кластеров в сетях второго уровня от количества людей, попавших в кластер базовой нейронной сети (первого уровня).

4.1 Набор данных о пользователях для исследования

В качестве выборки для обучения каскадной нейронной сети было собрано 90361 человек. Из-за ограничения ВКонтакте на получение данных о пользователях, а именно максимально возможное значение количества человек на запрос без параметров было 1000 человек, было решено собирать по 1000 человек из 1 города с возрастом от 18 до 60 лет.

Пример полученной и сохраненной информации можно увидеть в листинге 2.

4.2 Диаграмма Эндрюса

Одним из самых простых решений для визуализации n-мерного пространства является точка, спроецированная в двумерное или трехмерное пространство. Полученная размерность пространства составляет 18.

Суть диаграммы Эндрюса заключается в том, что каждая точка представляется в виде ряда Фурье:

$$f_x(t) = \frac{x_1}{\sqrt{2}} + x_2 \sin(t) + x_3 \cos(t) + x_4 \sin(2t) + x_5 \cos(2t) + ... \qquad (18)$$

Получившаяся функция изображается на графике на промежутке в пределах: $[-\pi, \pi]$

В качестве данных для построения было использовано математическое ожидания векторов, попавших в определенный кластер:

$$M(X) = \frac{\sum_1^n x_i}{n} \qquad (19)$$

Диаграмма Эндрюса, составленная из 5 различных выборок соответствующих кластеров представлена на рисунке 13.

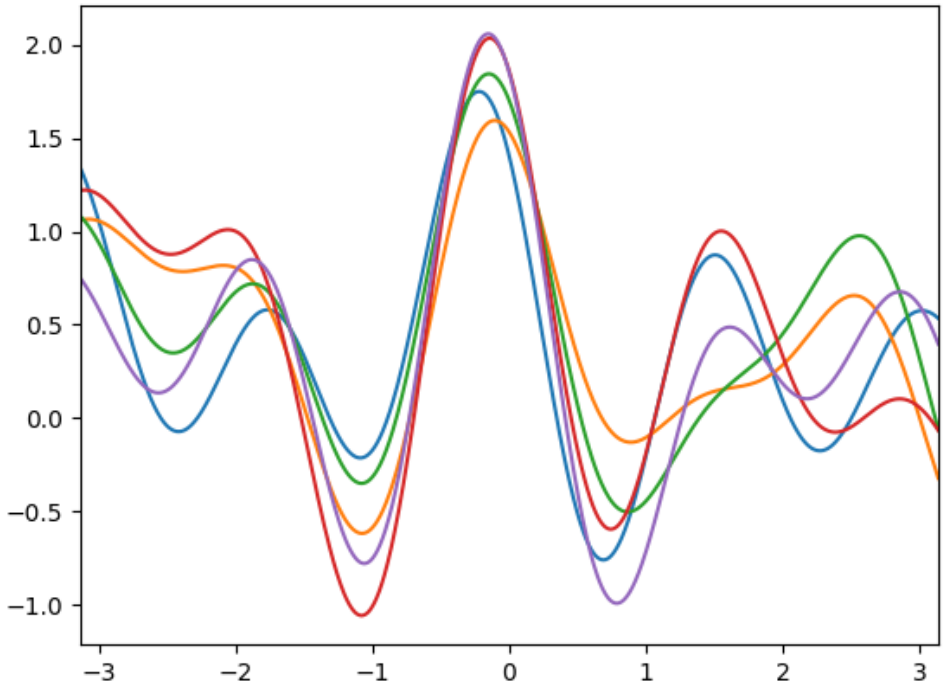

Рисунок 13 – Диаграмма Эндрюса

По данному графику можно увидеть, что математическое ожидание выборок разных кластеров не идентичны, что может свидетельствовать о корректном разбиении на кластеры.

4.3 Анализ выделенных кластеров

Для анализа выделенных кластеров в первую очередь необходимо найти математическое ожидание кластера:

$$x_{\text{ср}} = \frac{\sum_1^n x_i}{n} \tag{20}$$

Далее вычислить среднеквадратичное отклонение:

$$S = \sqrt{\frac{\sum_1^n (x_i - x_{\text{ср}})^2}{n}} \tag{21}$$

На рисунке 14 представлена гистограмма вычисленных среднеквадратич-

ных отклонений для всей выборки, кластеров сети Кохонена и кластеров каскадной сети Кохонена.

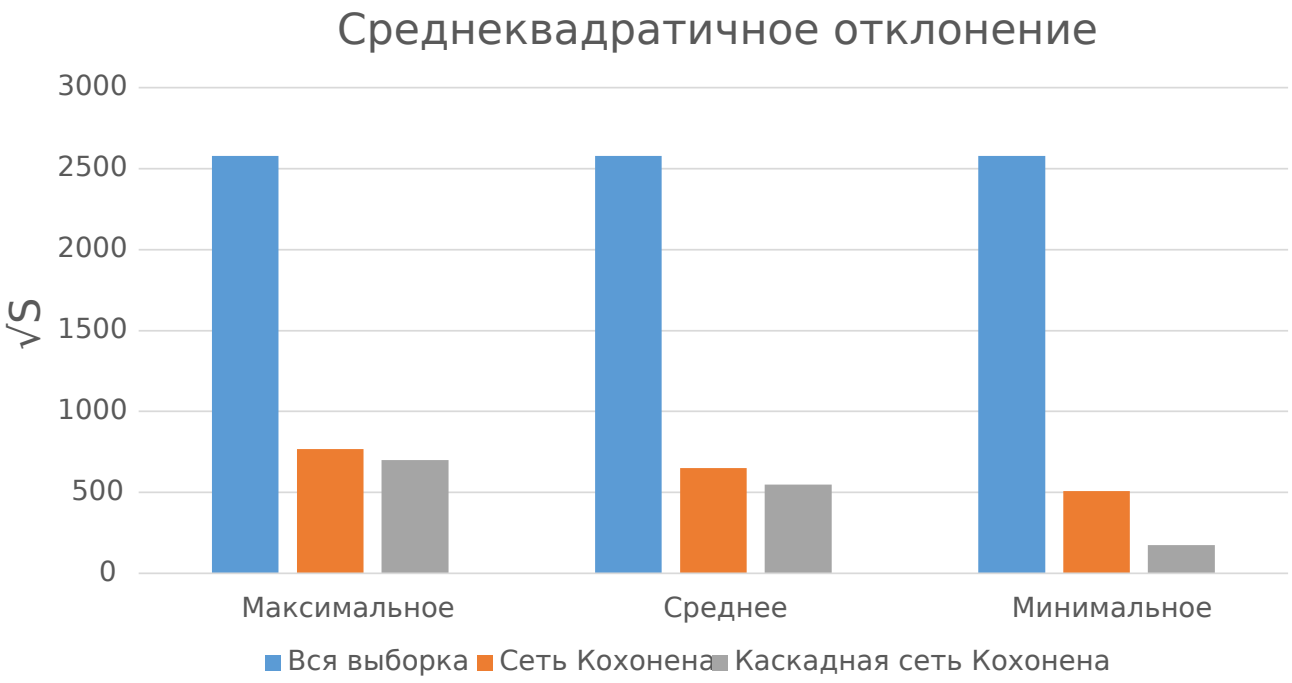

Рисунок 14 – Гистограмма вычисленных среднеквадратичных отклонений

На данной гистограмме можно увидеть, что среднеквадратичное отклонение всей выборки в разы превышает дисперсии выделенных кластеров.

Также можно сказать, что модификация позволила уменьшить среднеквадратичное отклонение выделенных кластеров, что свидетельствует о более точном разбиении на кластеры каскадной сети Кохонена.

4.4 Необходимость разделения кластеров на более мелкие

После обучения каскадной нейронной сети на 90361 человеке получилось, что общее число «живых» нейронов базовой нейронной сети Кохонена (первого уровня) стало равным 26. Начальное число кластеров базовой нейронной сети было равным 60. То есть процент мертвых нейронов составляет 44%.

После построения каскада общее число нейронов (выделенных кластеров) стало равным 44. Для сетей второго уровня начальное число нейронов было задано равным 20.

На рисунке 15 продемонстрирована зависимость количества выделенных кластеров в сетях второго уровня от среднеквадратичного отклонения кластера базовой нейронной сети.

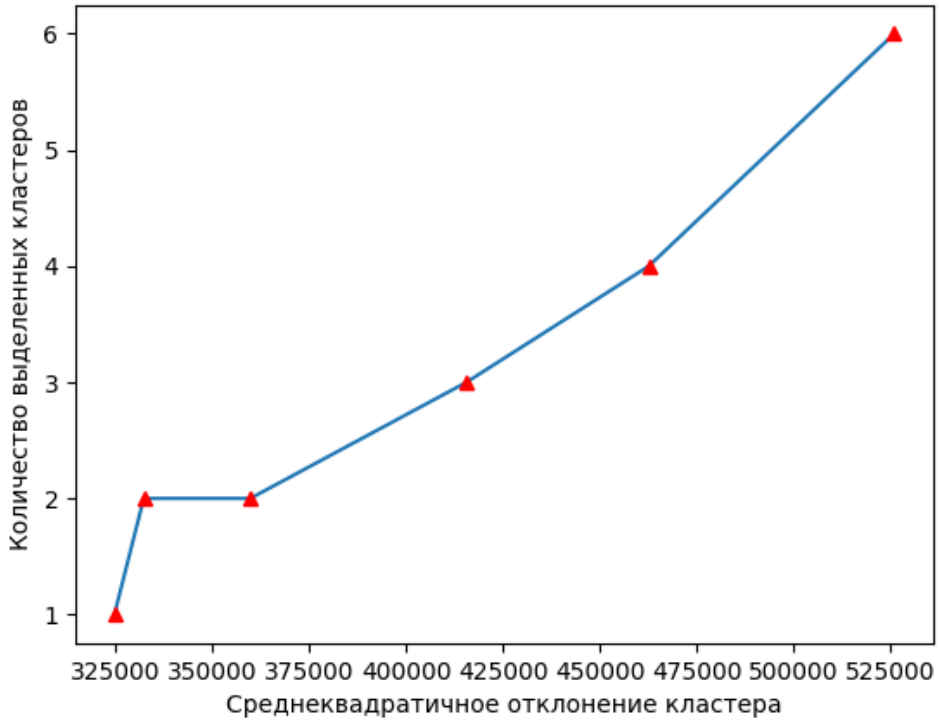

Рисунок 15 – Зависимость количества кластеров от среднеквадратичного отклонения

При проведении данного эксперимента изначально сохранялись индексы пользователей, попавших в кластер и при обучении сети второго уровня данные нормализовались конкретно для новой сети.

По графику можно увидеть, что чем больше количество людей в кластере, тем больше выделяется дополнительных кластеров.

4.5 Выводы

В данном разделе было описано на какой выборке проводилось обучение нейронной сети, проведены эксперименты с каскадной нейронной сетью Кохонена, показывающие корректное разбиение на кластеры, а также необходимость разбиения кластеров на более мелкие.

В результате исследований по графику Эндрюса можно сказать, что математическое ожидание выборок разных кластеров не идентичны, и это может свидетельствовать о корректном разбиении на кластеры.

Также можно сказать, что модификация позволила уменьшить среднеквадратичное отклонение выделенных кластеров, что свидетельствует о более точном разбиении на кластеры каскадной сети Кохонена.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы был разработан и реализован метод персонализации пользователей социальных сетей с использованием каскадной нейронной сети Кохонена, которая, после проведения исследования, показала высокую точность выделения кластеров.

Были решены следующие задачи:

1) проведен анализ существующих методов персонализации пользователей социальных сетей;
2) изучены способы получения информации о пользователях социальных сетей;
3) разработан метод персонализации пользователей социальных сетей на основе модифицированной нейронной сети;
4) разработано программное обеспечение, реализующее этот метод;
5) проведено исследование применимости разработанного обеспечения.

В качестве дальнейшего развития можно выделить следующие пункты:

1) увеличение размера выборки для обучения каскадной нейронной сети Кохонена;
2) увеличение количества информации, собираемой об одном человеке;
3) исследование пользователей, попавших в один и тот же кластер, предоставляя им примеры рекламы на одинаковую тематику.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Рульков, ВС. Нейронные сети в сфере интернет-маркетинга / ВС Рульков // Современные проблемы гуманитарных и естественных наук. — 2018. — Pp. 18–21.

2. Дмитриев, Егор Андреевич. Линейные классификаторы / Егор Андреевич Дмитриев. — 2017.

3. Гафаров, Фаиль Мубаракович. Искусственные нейронные сети и приложения. — 2018.

4. Галанов, АЭ. Нейронные сети и нейронные технологии / АЭ Галанов, ГП Селюкова // Актуальные вопросы науки и хозяйства: новые вызовы и решения. — 2019. — Pp. 399–405.

5. Созыкин, Андрей Владимирович. Обзор методов обучения глубоких нейронных сетей / Андрей Владимирович Созыкин // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. — 2017. — Vol. 6, no. 3.

6. Егоров, Александр Вадимович. Особенности методов кластеризации данных / Александр Вадимович Егоров, Наталия Игоревна Куприянова // Известия Южного федерального университета. Технические науки. — 2011. — Vol. 124, no. 11.

7. Sinaga, Kristina P. Unsupervised K-means clustering algorithm / Kristina P Sinaga, Miin-Shen Yang // IEEE access. — 2020. — Vol. 8. — Pp. 80716–80727.

8. Митрофанова, АС. Обучение перцептрона / АС Митрофанова, ГВ Комлев // Тенденции развития науки и образования. — 2019. — no. 49-12. — Pp. 69–71.

9. Ершов, КС. Анализ и классификация алгоритмов кластеризации / КС Ершов, ТН Романова // Новые информационные технологии в автоматизированных системах. — 2016. — no. 19.

10. Alhawarat, Mohammad. Revisiting K-means and topic modeling, a comparison study to cluster arabic documents / Mohammad Alhawarat, M Hegazi // IEEE Access. — 2018. — Vol. 6. — Pp. 42740–42749.

11. Scalable k-means++ / Bahman Bahmani, Benjamin Moseley, Andrea Vattani et al. // arXiv preprint arXiv:1203.6402. — 2012.

12. Габдрахманова, НТ. Кластеризация документов с помощью нейронных сетей / НТ Габдрахманова // Речевые технологии. — 2019. — no. 1. — Pp. 45–53.

13. Maugis-Rabusseau, Cathy. Adaptive density estimation for clustering with Gaussian mixtures / Cathy Maugis-Rabusseau, Bertrand Michel // ESAIM: Probability and Statistics. — 2013. — Vol. 17. — Pp. 698–724.

14. Schieferdecker, Dennis. Gaussian mixture reduction via clustering / Dennis Schieferdecker, Marco F Huber // 2009 12th international conference on information fusion / IEEE. — 2009. — Pp. 1536–1543.

15. Du, K-L. Clustering: A neural network approach / K-L Du // Neural networks. — 2010. — Vol. 23, no. 1. — Pp. 89–107.

16. Горбаченко, ВИ. Сети и карты Кохонена / ВИ Горбаченко // Научноисследовательский центр самоорганизации и развития систем.–2010.–Режим доступа: http://gorbachenko. self-organization. ru. — 2010.

17. Мамаев, Иван Иванович. Применение карт Кохонена для анализа основных социально-экономических показателей административных

районов Ставропольского края / Иван Иванович Мамаев, Павел Анатольевич Сахнюк, Татьяна Ивановна Сахнюк // Russian Journal of Education and Psychology. — 2012. — no. 12.

18. Nizam, Muhammad. Kohonen neural network clustering for voltage control in power systems / Muhammad Nizam // Telkomnika. — 2010. — Vol. 8, no. 2. — P. 115.

19. Ettaouil, Mohamed. Architecture optimization model for the multilayer perceptron and clustering. / Mohamed Ettaouil, Mohamed Lazaar, Youssef Ghanou // Journal of Theoretical & Applied Information Technology. — 2013. — Vol. 47, no. 1.

ПРИЛОЖЕНИЕ А

Пример ответа на запрос users.get


```
{
    "response": [
        {
            "id": 40409863,
            "first_name": "Denis",
            "last_name": "Sklifasovskiy",
            "can_access_closed": true,
            "is_closed": false,
            "sex": 2,
            "online": 1,
            "bdate": "4.9.2000",
            "city": {
                "id": 1,
                "title": "Moscow"
            },
            "country": {
                "id": 1,
                "title": "Russia"
            },
            "timezone": 3,
            "has_photo": 1,
            "has_mobile": 1,
            "activities": "",
            "followers_count": 228,
            "career": [],
            "military": [
                {
                    "country_id": 1,
                    "unit": "  ",
                    "unit_id": 227
```

```
            }
        ],
        "university": 250,
        "university_name": "BMSTU",
        "faculty": 0,
        "faculty_name": "",
        "graduation": 0,
        "home_town": "Moscow",
        "relation": 2,
        "personal": {
            "alcohol": 0,
            "inspired_by": "",
            "langs": [
                "Russian"
            ],
            "life_main": 0,
            "people_main": 0,
            "smoking": 1
        },
        "counters": {
            "albums": 0,
            "audios": 979,
            "followers": 228,
            "friends": 165,
            "gifts": 142,
            "groups": 199,
            "online_friends": 15,
            "pages": 124,
            "photos": 24,
            "subscriptions": 0,
            "user_photos": 0,
            "videos": 18,
            "new_photo_tags": 0,
            "new_recognition_tags": 0,
            "clips_followers": 393
```

```
66                  }
67              }
68          ]
69      }
```