**СОДЕРЖАНИЕ**

# ВВЕДЕНИЕ

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

Социальная сеть - это платформа, которую формируют несколько человек. Люди делятся контентом, новостями, рекламой и другой информацией. Обмен всеми видами информации может осуществляться для информирования всех в различных целях. Эти цели могут быть разного типа, например, реклама и продажа товаров, социальные связи, хобби, работа и т.д.

Для достижения этих целей может оказаться очень полезным формирование групп с одинаковыми желаниями. Каждый, у кого есть друзья и родственники в социальных сетях, а также знакомые подписчики, может создать группу.

Кроме того, социальные сети являются популярной платформой для рекламы. Владельцы групп могут использовать их для продвижения своих товаров или услуг. Это может быть как маленький бизнес, который хочет привлечь новых клиентов, так и крупная компания, рекламирующая свои продукты или проводящая акции.

Таргетированная реклама и кластеризация занимают прочные позиции в области продвижения различных товаров. Данная отрасль развивается быстрыми темпами и важно успеть отслеживать изменения в этой 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 огромном количестве этой информации, чтобы найти наиболее подходящий им продукт или услугу.

Во-вторых, понимание различных 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.

Например, если компания проводит рекламную кампанию на продажу спортивной одежды, она может использовать кластеризацию, чтобы выделить группы пользователей, интересующихся спортом или фитнесом. Затем компания может настроить свои рекламные объявления таким образом, чтобы они были более релевантными для каждой группы, повышая эффективность рекламной кампании и увеличивая вероятность привлечения потенциальных клиентов.

В последнее время для точного изучения предпочтений пользователей и атрибутов товаров широко используются модели, основанные на машинном обучении. Преимущество машинного обучения в том, что оно может точно фиксировать представления о пользователях.

Применение нейронной сети в маркетинговой деятельности позволит выдавать наиболее подходящие товары, рекламные продукты, услуги непосредственному клиенту, что позволит повысить 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 кластеризации пользователей социальных сетей с использованием нейронных сетей для обеспечения таргетированной рекламы.

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 кластеризации пользователей социальных сетей на основе модифицированной нейронной сети;
4) разработать программное обеспечение, реализующее этот метод;
5) провести исследование применимости разработанного программного обеспечения.

# Глава 1. Методы кластеризации пользователей

## 1.1. Постановка задачи

Рассмотрим задачу кластеризации пользователей социальных сетей для таргетированной рекламы с использованием нейронных сетей. На вход алгоритма кластеризации подается индентификатор пользователя социальной сети. Выходом алгоритма является номер кластера, или же индекс таргетированной рекламы, которая интересна пользователю. Таким образом, задача кластеризации 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 областей бизнеса. У большинства социальных сетей есть собственные алгоритмы кластеризации пользователей для предоставления рекламы различных товаров или услуг.

### 1.2.1. Яндекс Директ

Яндекс делит своих пользователей на различные сегменты на основе собранной о них информации. Один пользователь по различным признакам может относиться к различным сегментам. Каждый год количество анализируемых метрик непрерывно растет. Если о пользователе еще нет информации, то

реклама показывается в зависимости от наполнения страницы, которую посетил пользователь.

### 1.2.2. VK

Социальная сеть VK 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. Дзен

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.

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.

### 1.2.4. Вывод

Все рассмотренные варианты имеют 2 самых главных недостатков:

1) техническая реализация предложенных решений находится в закрытом доступе и нельзя точно определить какие алгоритмы машинного обучения для кластеризации они используют;
2) невозможно точно определить для каких конкретных целей используется каждая из технологий;

Таким образом, существует потребность в системах кластеризации пользователей социальных сетей.

## 1.3. Выбор социальной сети для получения информации о пользователях

Для формирования большой 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. Мессенджер Telegram

Мессенджер Telegram появился в 2013 году.

На данный момент количество активных пользователей составляет 900 млн в месяц.

У телеграмма есть открытое апи для создания чат-ботов, но он так и не предоставляет способ сбора информации о пользователях или подписчиков определенных каналов.

### 1.3.3. 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) [1] – 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 [2].

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 выбирать центры один за другим контролируемым образом, где текущий набор выбранных центров будет стохастически смещать выбор следующего центра [3].

Вычисление цетроида:

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

Необходимо определить стоимость $Y$ по отношению к $C$ как:

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

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 [4] – 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 C-средних (FCM). Он основан на концепции нечеткого C-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 C 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 [5]. 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 [6]. 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:

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

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

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

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

### 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 [7]. 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 обучением. 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 конкурентоспо-

собная 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 [8]. 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).

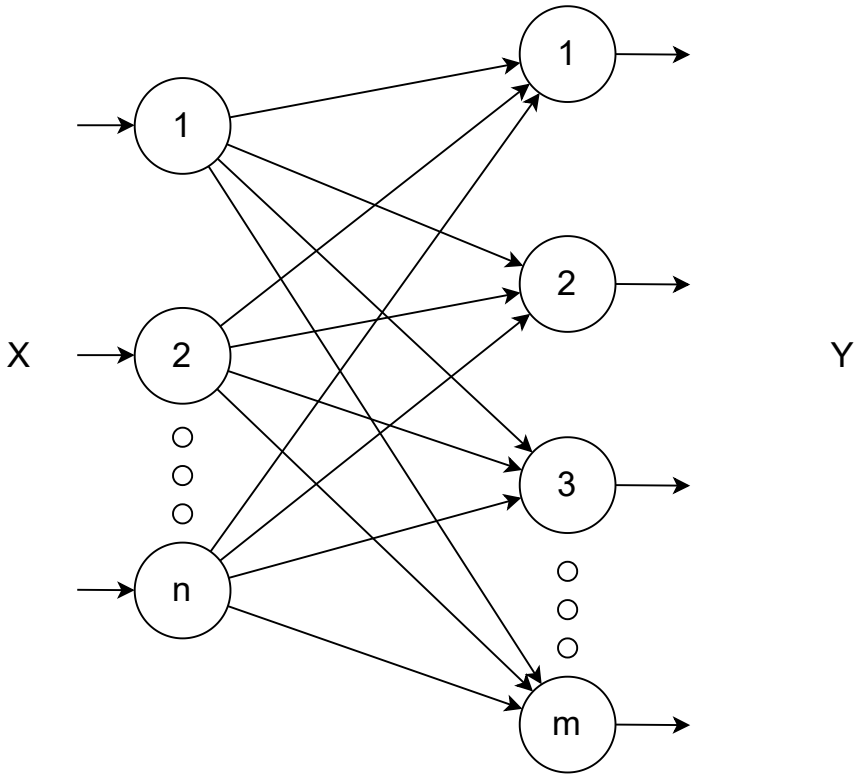


Рисунок 2 – Структура сети Кохонена[9]

Для обучения сети применяются механизмы конкуренции. Перед процессом обучения производится инициализация сети, то есть первоначальное зада-

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

Алгоритм учитывает евклидово расстояние между двумя n-мерными векторами, которое измеряется сходством между входными векторами [10]. 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

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

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 на-

ходится 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.

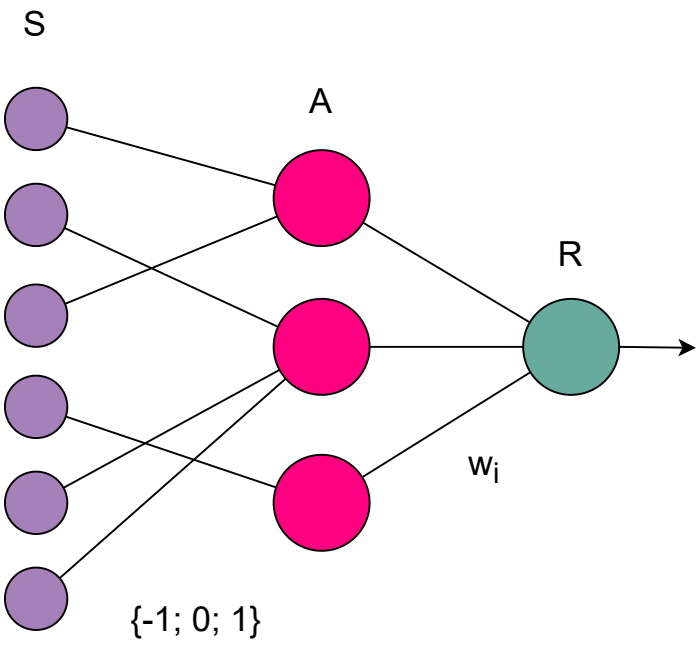


Рисунок 3 – Схема перцептрона

Кроме классического метода обучения перцептрона, Розенблат также ввел понятие об обучении без учителя, предложив следующий способ обучения: альфа-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 [11]. 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, 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.

# Глава 2. Разработка подхода кластеризации пользователей социальных сетей

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.

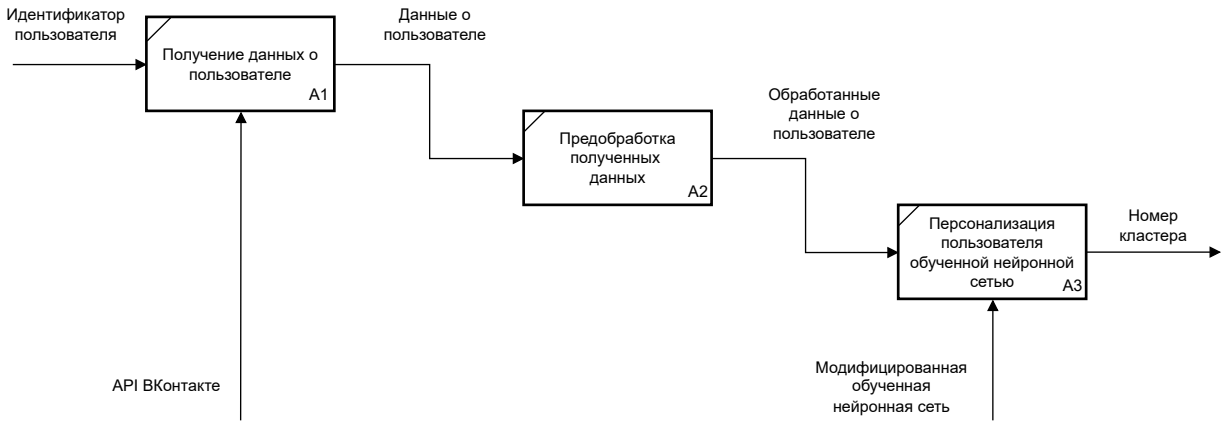


Рисунок 4 – IDEF0 диаграмма второго уровня

Результат работы программы - номер кластера для таргетированной рекламы выбранного пользователя.

## 2.2. Получение информации о пользователях социальных сетей

API VK является бесплатным для разработчиков. Это интерфейс, который позволяет получать множество различных данных непосредственно из баз дан-

ных социальной сети. Синтаксис запросов и тип возвращаемых данных определены на стороне сервиса.

Для использования этого API необходимо получить специальный ключ доступа. Для получения ключа необходимо зарегестрироваться в VK и создать standalone-приложение [12].

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

Для сбора информации о пользователях понадобятся следующие методы:

1) users.search – возвращает список объектов, описывающих 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.

Например, если пользователь подписан в большинстве на новостные группы, то ему проставляется соответствующий идентификатор по жанру новостей.

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

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

В нем представлены следующие поля:

1) bdate – дата рождения;
2) career – информацию о карьере;
3) city – город проживания;
4) country – страну проживания;
5) followers_count – количество подписчиков;
6) education – информацию об образовании;

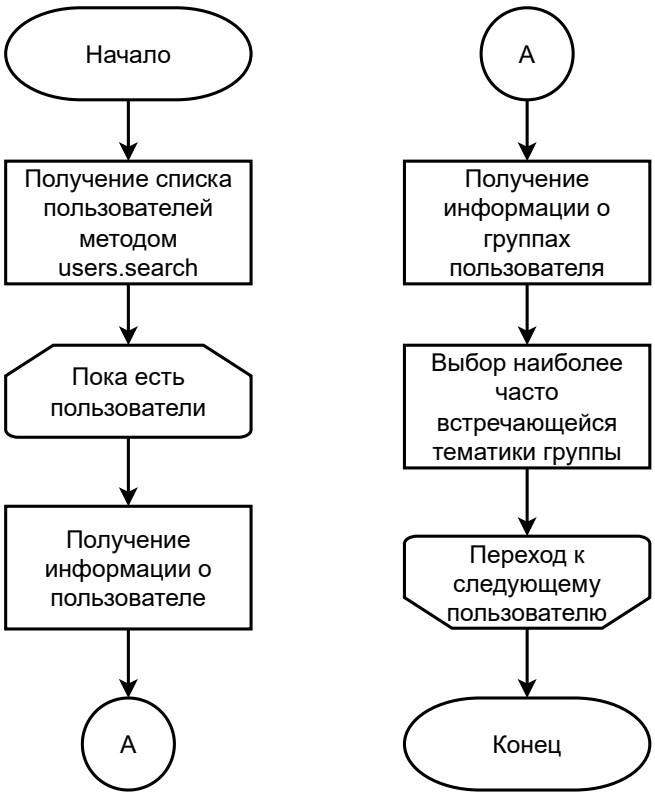


Рисунок 5 – Алгоритм получения информации о пользователях

7) has_mobile – есть ли телефон;
8) has_photo – есть ли фотографии;
9) counters – различные счетчики (количество друзей, групп, страниц);
10) home_town – город рождения;
11) military – информации о прохождении службы;
12) personal – различную информацию о жизненной позиции;
13) relation – информация об отношениях;
14) sex – пол человека;
15) timezone – временная зона.

### 2.3. Хранение информации о пользователях социальных сетей

Существует несколько независимых инструментов системы управления базами данных для работы большими объемами данных, однако большинство из этих инструментов недоступны организациям, поскольку они очень дорогостоящие [13].

Реляционная база данных - это совокупность нескольких таблиц [14]. Эти

таблицы связаны с другими таблицами с помощью связей. База данных хранит набор данных. Как правило, в базах данных важно сохранять, а также считывать их при необходимости.

Хранение данных является важным элементом любой системы. Исторически сложилось так, что существует несколько популярных вариантов, таких как файловые, иерархические, документные и, что более важно, реляционные базы данных.

Поскольку существует множество типов баз данных, которые можно выбрать для конкретного решения, реляционная база данных имеет множество преимуществ перед другими базами данных, такими как файловые, документные и иерархические базы данных. Основной возможностью реляционных баз данных является возможность создания связей между таблицами. Каждая таблица состоит из столбцов и строк.

Для хранения информации о пользователях социальных сетей наиболее подходящей является реляционная база данных.

Многие компании сейчас переходят на программное обеспечение с открытым исходным кодом. Одной из таких СУБД является PostgreSQL, поскольку это самая передовая СУБД с открытым исходным кодом в мире, она поддерживает большинство транзакций SQL, дает возможность использовать сложные запросы, триггеры, представления, целостность транзакций и позволяет добавлять данные расширения типов, функции, операторы и процедурные языки.

На рисунке 6 прдеставлена таблица User, в которой будут храниться собранные данные о юзере.

### 2.4. Формирование обучающей выборки

Обучение нейронной сети является важным этапом ее работы.

Перед началом процедуры обучения особое внимание уделяется предварительной обработке данных [15]. Большинство исследований, посвященных применению нейронных сетей, сводят процедуры предварительной обработки к нормализации, масштабированию и предварительной инициализации весов.

| User | |
|---|---|
| PK | id |
| | can_access_closed |
| | sex |
| | online |
| | city |
| | country |
| | has_photo |
| | has_mobile |
| | followers_count |
| | career |
| | university |
| | faculty |
| | graduation |
| | relation |
| | political |
| | people_main |
| | life_main |
| | smoking |
| | alcohol |
| | activity |

Рисунок 6 – Таблица User

Для эффективного обучения нейронной сети следует учитывать особенности распределения исходных данных, что может быть слишком сложным из-за большого количества факторов. В этом случае целесообразно использовать кластеризацию для формирования обучающего набора из примеров атрибутов, которые являются наиболее уникальными в наборе.

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 7.

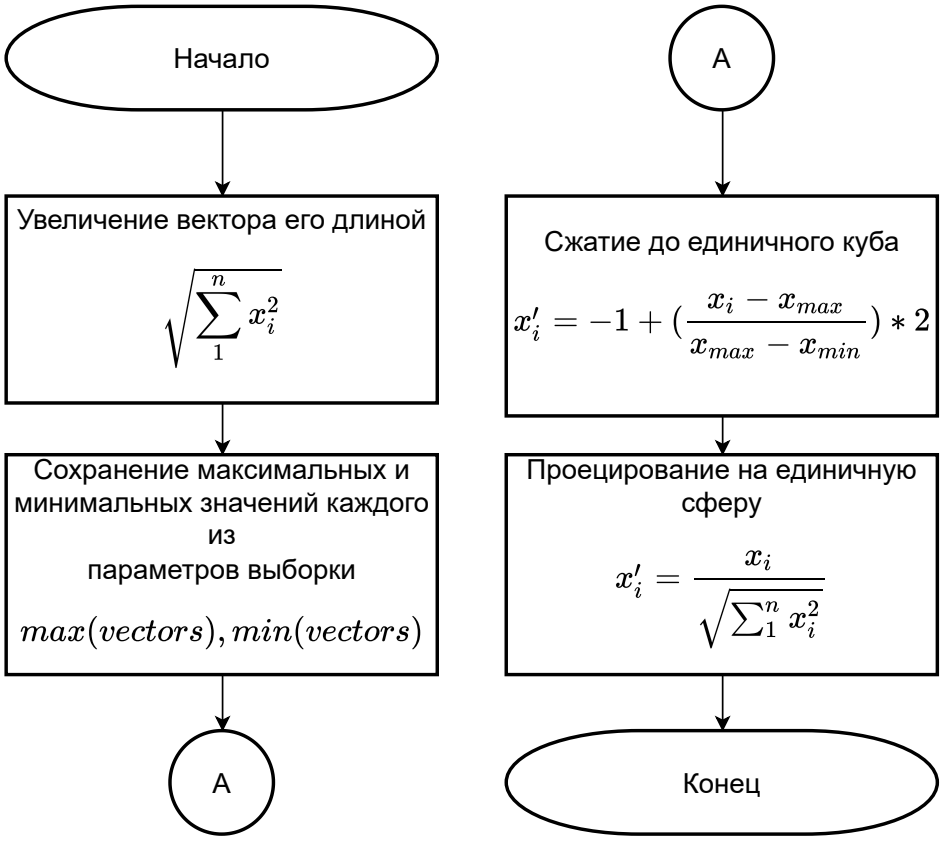


Рисунок 7 – Схема алгоритма формирования обучающей выборки

## 2.5. Архитектура нейронной сети Кохонена

На рисунке 8 представлена архитектура нейронной сети Кохонена.

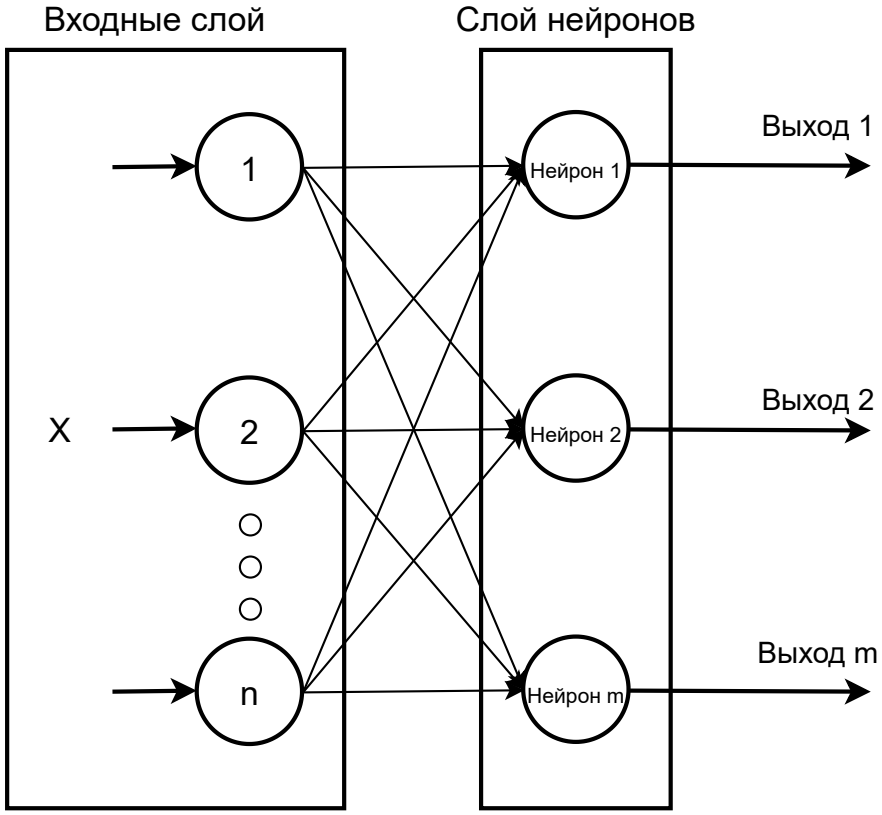


Рисунок 8 – Архитектура нейронной сети Кохонена

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.

### 2.6. Обучение нейронной сети

В нейронной сети Кохонена вес представляет компоненты каждого центра кластера, а количество узлов в выходном слое представляет количество кластеров [16].

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

#### 2.6.1. 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, $\alpha$ – 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.

### 2.6.2. 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 9 представлена схема алгоритма обучения нейронной сети.

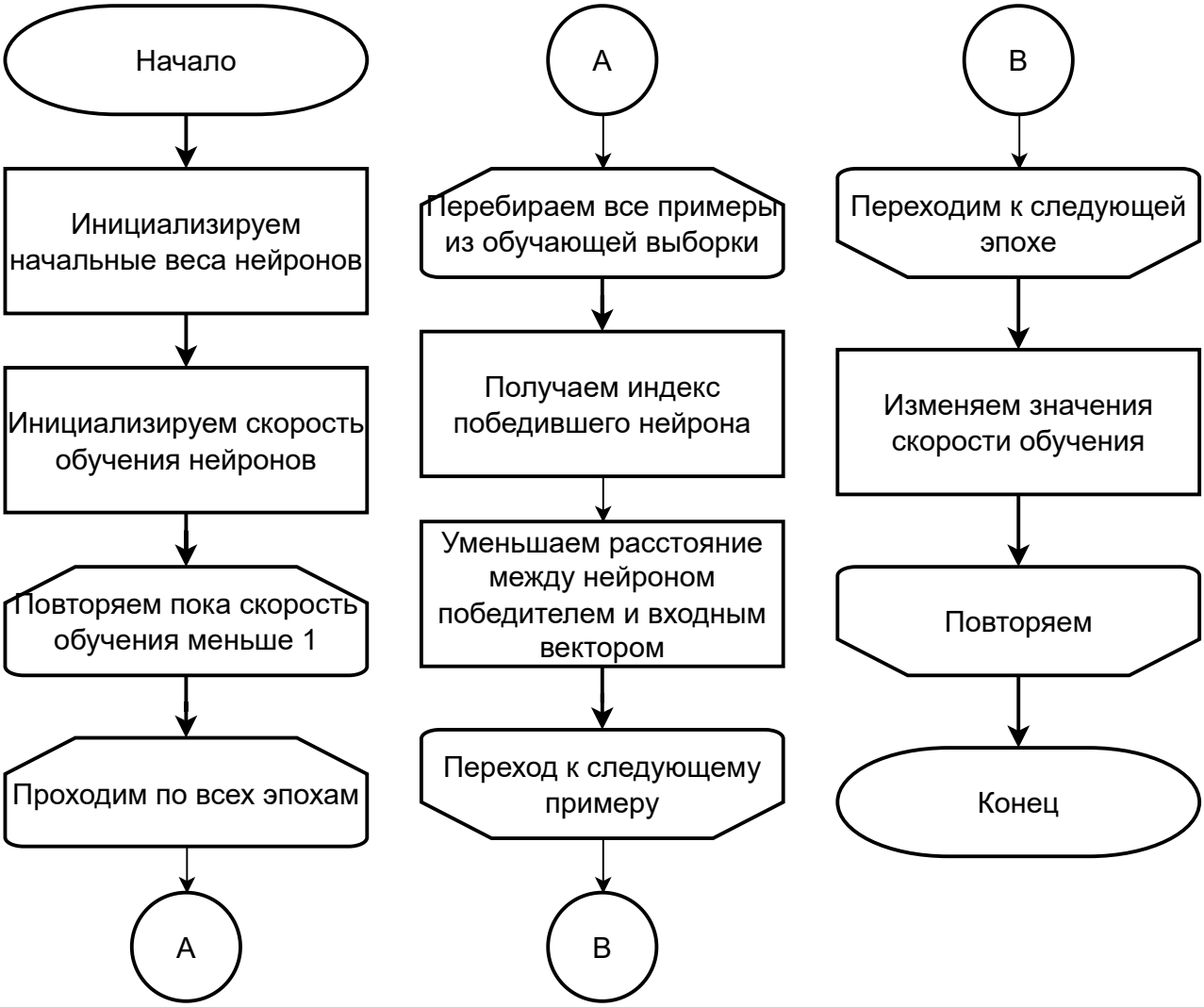


Рисунок 9 – Схема алгоритма обучения

### 2.6.3. 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 10 представлена схема алгоритма поиска нейрона победителя.

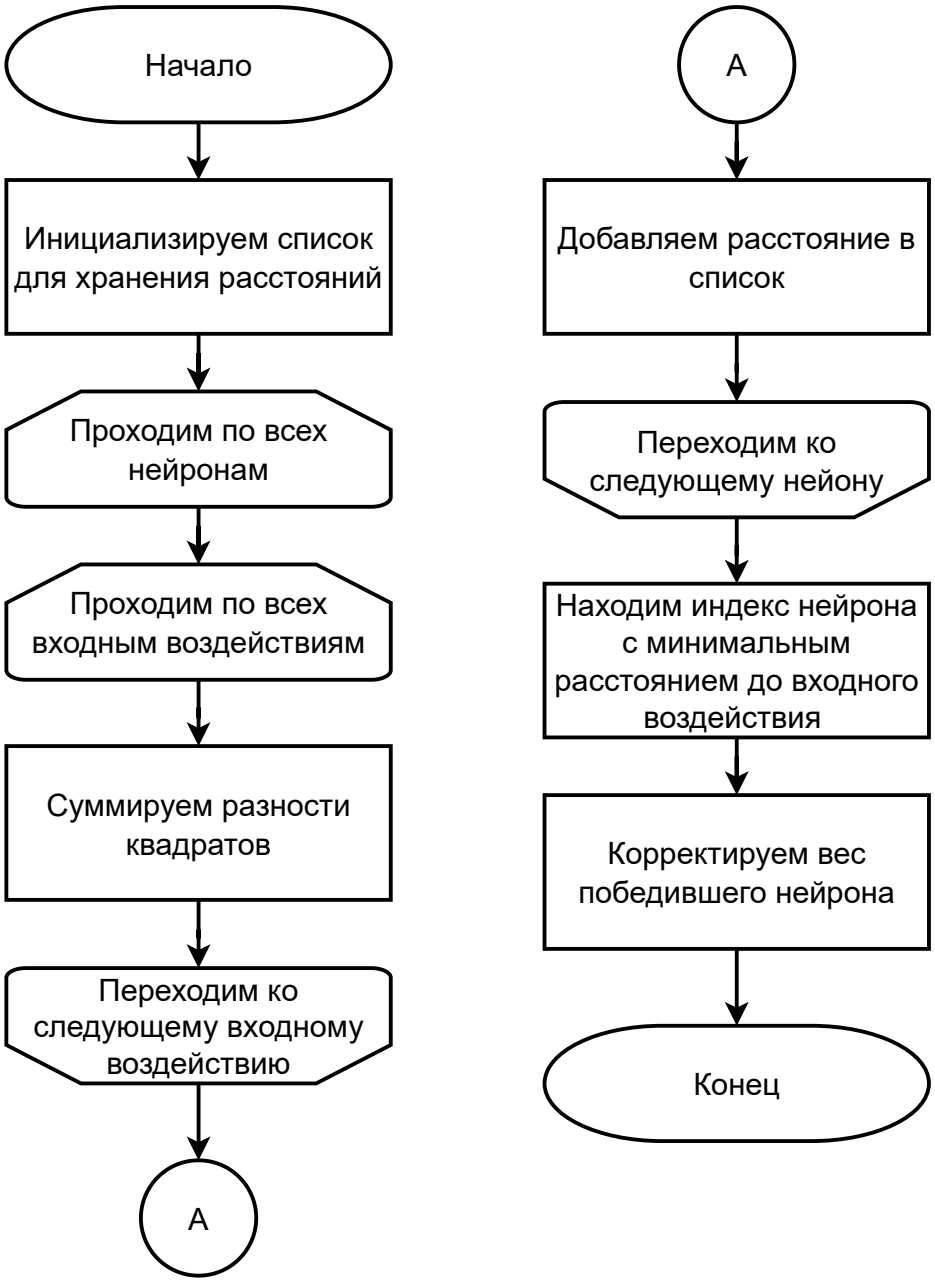


Рисунок 10 – Схема алгоритма поиска нейрона «победителя»

## 2.7. 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 11 представлена каскадная нейронная сеть Кохонена.

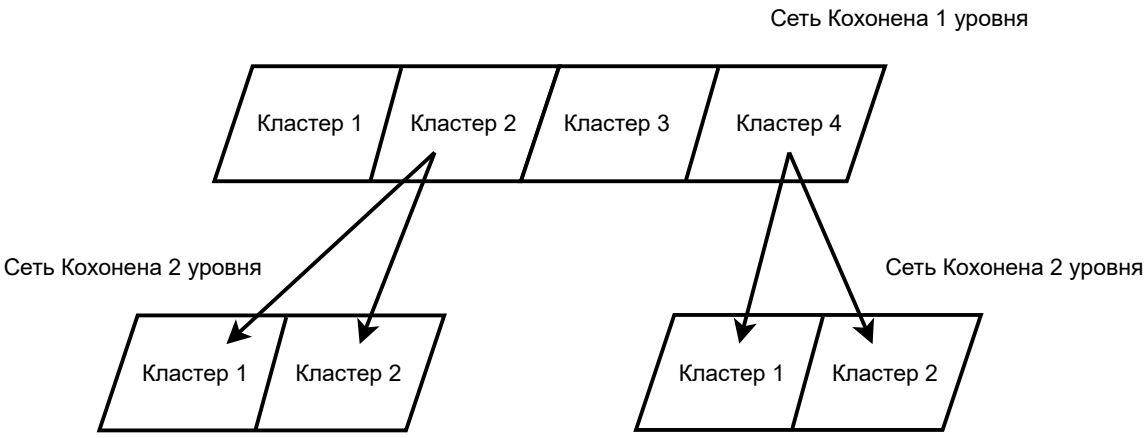


Рисунок 11 – Схема алгоритма поиска нейрона «победителя»

## 2.8. 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.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ


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

2. 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.

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

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

5. 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.

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

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

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

9. 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.

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

11. 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.

12. VK. Быстрый старт. — https://dev.vk.com/ru/api/getting-started. — 2024. — Accessed: 2024-10-27.

13. Integration of data mining techniques to PostgreSQL database manager system / Amelec Viloria, Genesis Camargo Acuña, Daniel Jesús Alcázar Franco et al. // Procedia Computer Science. — 2019. — Vol. 155. — Pp. 575–580.

14. Asanka, PPG. Database Design and Modeling with PostgreSQL / PPG Asanka. — 2020.

15. Pastukhov, Aleksey A. Kohonen self-organizing map application to representative sample formation in the training of the multilayer perceptron / Aleksey A Pastukhov, Alexander A Prokofiev // St. Petersburg Polytechnical University Journal: Physics and Mathematics. — 2016. — Vol. 2, no. 2. — Pp. 134–143.

16. Mao, Youwen. Application of Kohonen Neural Network in Sports Cluster /

Youwen Mao // Wireless Communications and Mobile Computing. — 2022. — Vol. 2022, no. 1. — P. 2266702.

# 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
        }
```

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