Е.А. Валяев, З.З. Мингалиев

НЕЙРОСЕТЕВОЙ КАСКАД НА ОСНОВЕ САМООРГАНИЗУЮЩИХСЯ КАРТ КОХОНЕНА

(Казанский национальный исследовательский технический университет – КАИ им. А.Н. Туполева)

1 Введение

Люди всегда пытались классифицировать живые и неживые объекты, которые их окружают. Объединение объектов в группы является необходимым условием для их наименования. Это требует распознавание дискретных подмножеств в окружающей среде, которая чаще всего непрерывна.

Кластеризовать - это значит признать, что объекты достаточно похожи, чтобы быть помещенными в одну и ту же группу. Существуют различные критерии, которые могут использоваться, для определения того, являются ли объекты достаточно похожими для выделения в группу.

Кластеризация – это операция многомерного анализа, которая заключается в разбиении коллекции объектов (или дескрипторов) в исследовании. Классификация объектов (или дескрипторов) в результате кластеризации может включать одну секцию или несколько иерархически вложенных разделов объектов (или дескрипторов) в зависимости от выбранной модели кластеризации.

Кластеризация - важная часть экологических исследования и экологии в целом уже долгое время. Часто в результате получения результатов наблюдения и экспериментов мы ничего не знаем о внутренней структуре наблюдаемых явлений и все, что у нас есть это некоторая совокупность данных. В данном случае следует обнаружить такую структуру, которая соответствовала бы наблюдениям, т.е. найти «естественную группу», а также и подбор подходящего смысла для этого термина.

2 Постановка задачи

Создать вычислительный алгоритм для возможности кластеризации и представления получившихся кластеров в графическом виде с произвольной точностью.

3 Метод

Для получения наиболее точных результатов мы будем использовать модель нейросетевого каскада, который основан на использовании SOM-карт Кохонена. В отличие от стандартной нейронной сети Кохонена, каскад представляет собой последовательность нескольких кластеризующих нейронных сетей- слоев каскада. Каждый следующий слой осуществляет кластеризацию с более высокой степенью детализации, чем предыдущий. Необходимое число последовательных кластеризующих слоев определяется по

мере анализа результатов кластеризации предшествующим слоем.

На первом этапе происходит формирование SOM -карты для всего исходного набора данных. После кластеризации и анализа степени однородности данных в каждом получившимся кластере, мы можем сформировать следующий слой SOM применительно к той группе, чья однородность нас по каким-либо причинам не устроила. Процесс продолжается до тех пор, пока не будут получены приемлемые для нас результаты кластеризации по всем выделенным кластерам.

Основной целью карт самоорганизации (self-organizing map - SOM) является изменение входящих сигналов с произвольной размерностью в одно - или двухмерную дискретную карту. На рис.1 представлена двумерная решетка нейронов, которая используется в качестве дискретной карты. Все нейроны этой решетки осуществляют связь каждым узлом входного слоя.

Каждый входной слой состоит из локализованной области активности, размещаемой в относительно спокойной области. Расположение и природа такой области обычно варьируется в зависимости от конкретной реализации входного примера. Из этого следует, что для правильного развития процесса самоорганизации требуется, чтобы все нейроны сети были обеспечены достаточным количеством различных реализаций входных образов.

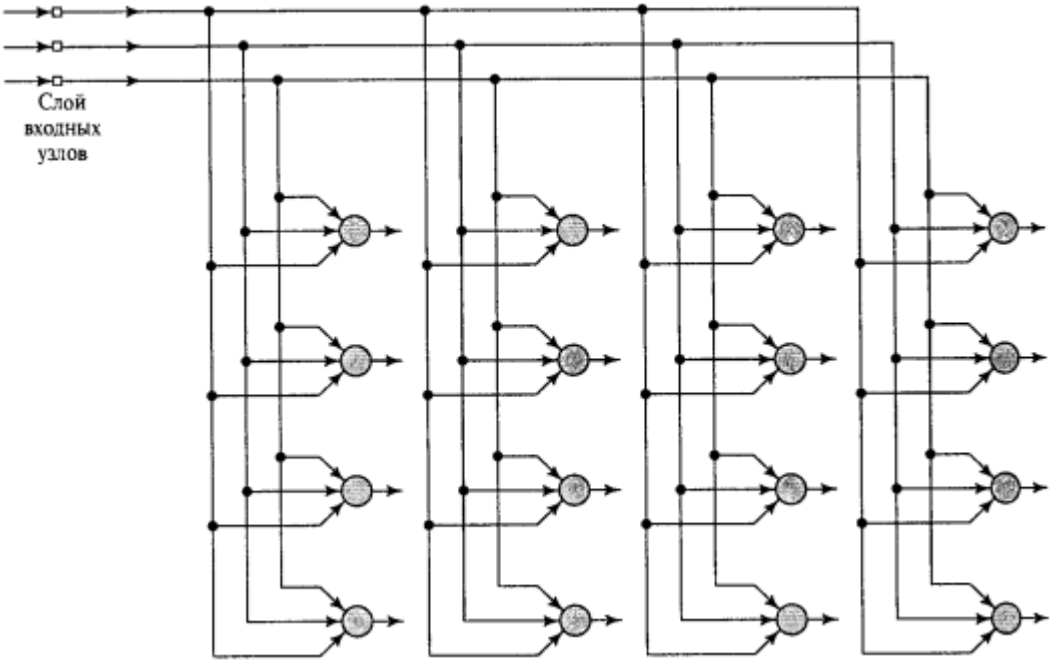

Рис. 1. Двумерная решетка нейронов

Алгоритм, формирующий карту самоорганизации, начинается с инициализации синаптических весов сети. Обычно это происходит с помощью назначения синаптическим весам малых значений, сформированных генератором случайных чисел. При таком формировании карта признаков изначально не имеет какого-либо порядка признаков. После корректной инициализации сети для формирования карты самоорганизации запускаются

три следующих основных процесса.

1. Конкуренция (competition). Для каждого входящего образа нейроны сети вычисляют относительные значения дискриминантной функции, тем самым осуществляя ту самую конкуренцию среди нейронов.

2. Кооперация (cooperation). Победивший нейрон формирует положение пространства топологической окрестности нейронов, таким образом между нейронами создается базис для кооперации.

3. Синаптическая адаптация (synaptic adaptation). Значения дискриминантных функций возбужденных нейронов увеличиваются относительно к входным образам с помощью корректировки синаптических весов. Изменения происходят так, чтобы отклик нейрона-победителя на последующее применение аналогичных примеров усиливался.

Идея алгортима Кохонена состоит в простом геометрическом вычислении свойств Хеббоподобного правила обучения и латеральных взаимодействий. Особенности данного алгоритма:

Непрерывное входное пространство образов активации, которые генерируются в соответствии с некоторым распределением вероятности.

Топология в форме решетки нейронов, которая определяет дискретное выходное пространство.

Функция окрестности $h_{j,i(x)}(n)$, определенная в окрестности нейрона-победителя $i(x)$ и зависящая от времени.

Параметр скорости обучения $\eta(n)$ с заданным начальным значением η_0, и который убывает во времени n, но при этом никогда не достигает нуля.

Для функции окрестности на этапе упорядочивания (т.е. приблизительно для первой тысячи итераций) можно использовать формулу

$$h_{j,i(x)}(n)= \exp(\frac{-d_{j,i}^2}{2\sigma^2(n)}), n = 0,1,2 \ldots,$$

где$\sigma(n) = \sigma_0 \exp(\frac{-n}{\tau_1}), n = 0,1,2 \ldots,$

где σ_0- начальное значение величины σв алгоритме SOM; τ_1 некоторая временная константа.

Для и параметра скорости обучения на этапе упорядочивания используем формулу

$$\eta(n) = \eta_0 \exp(\frac{-n}{\tau_2}), n = 0,1,2 \ldots,$$

где τ_2- временная константа алгоритма SOM.

Алгоритм проходит три основных шага: подвыборка (sampling), поиск максимального соответствия (similarity matching) и корректировка (updating).

Кратко описание всех шагов алгоритма:

1. Инициализация. Для исходных векторов синаптических весов $w_j(0)$выбираются случайные значения. Единственным требованием здесь является различие векторов для разных значений j =1,2,…,l, где l - общее количество нейронов в решетке. Еще одним способом инициализации алгоритма является случайный выбор множества весов $\{w_j(0)\}_{j=1}^{l}$из доступного

множества входных векторов $\{x_i\}_{i=1}^{N}$

2. Подвыборка. Осуществляется выбор вектора x из входного пространства с определенной вероятностью. Этот вектор x с размерностью т представляет собой возбуждение, которое применяется к решетке нейронов.

3. Поиск максимального подобия. Находится самый подходящий (победивший) нейрон $i(x)$на шаге n, используя критерий минимума Евклидова расстояния:

$$i(x) = arg\min_{j} \|x - w_j\|, j = 1,2, \dots, l.$$

4. Коррекция. Производим корректировку векторов синаптических весов всех нейронов, используя следующую формулу:

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n)(x - w_j(n)),$$

где $\eta(n) -$ параметр скорости обучения; $h_{j,i(x)}(n)$- функция окрестности с центром в победившем нейроне $i(x)$. Данные параметры во время обучения динамически изменяются для получения наилучшего результата.

5. Возврат к второму шагу и продолжение вычислений, пока в карте признаков не перестанут осуществляться значительные изменения.

После построения самоорганизующейся карты, для выделения кластеров мы используем метод k-средних. Преимущество данного метода заключается в том, что он легко реализуем на практике и имеет линейную сложность, т.е. быстр. Недостаток заключается в том, что для возможности использования этого метода необходимо иметь гипотезу о наиболее вероятном количестве кластеров.

4 Программная реализация.

Алгоритм реализован на высокоуровневом языке программирования Python.

В качестве исходных данных использован набор данных [1], в котором представлены измерения PM-2.5 (мелкодисперсных взвешенных частиц пыли в атмосферном воздухе), а также температуры, давления и др. в пяти разных китайских городах с 1 января 2014 года по 31 декабря 2015 года. Так как в некоторые дни по определенным измерениям нет данных, мы не включаем их в наше исследование. Таким образом, суммарное количество строк в наборе составило 100684.

Чтобы правильно отобразить карту с данными разных систем измерения, нужно масштабировать их относительно друг друга в единую систему счисления. Для этого мы используем метод MinMaxScaler из библиотеки Scikit-learn. После данного преобразования, значения всех признаков будут принадлежать отрезку от нуля до единицы.

Построение SOM производилось с помощью библиотеки Somoclu [2].

Размер SOM зависит от количества строк в исследуемом наборе данных. В статье "Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm" [3] эмпирическим путем была выведена закономерность

$$M \approx 5\sqrt{N}, \quad (1)$$

где M - количество нейронов, N - количество строк. Таким образом, мы рассчитали, что размер нашей карты будет 40x40 нейронов.

На этапе самоорганизации мы устанавливаем параметр скорости 0.1. Он будет экспоненциально убывать до 0.035. Функция окрестности должна охватывать почти все нейроны в сети (в нашем случае ее значение будет равно 38) и будет экспоненциально сужаться до 1. Количество итераций 1000. На выходе мы получаем карту (Рис.2), где цветом представлены значения нейронов.

На этапе сходимости количество итераций 15000, параметр скорости обучения будет меняться экспоненциально от 0.035 до 0.00 1, значение функции окрестности меняется экспоненциально от 1 до 0.000001.

Для выявления оптимального количества кластеров, мы используем Elbow method, который был импортирован из библиотеки Scikit-learn [4]. (Рис.3)

Проанализировав значения дисперсии для каждого набора кластеров, мы можем выбрать k=4, где k – количество кластеров, так как дальнейшее увеличение не даст существенного изменения дисперсии.

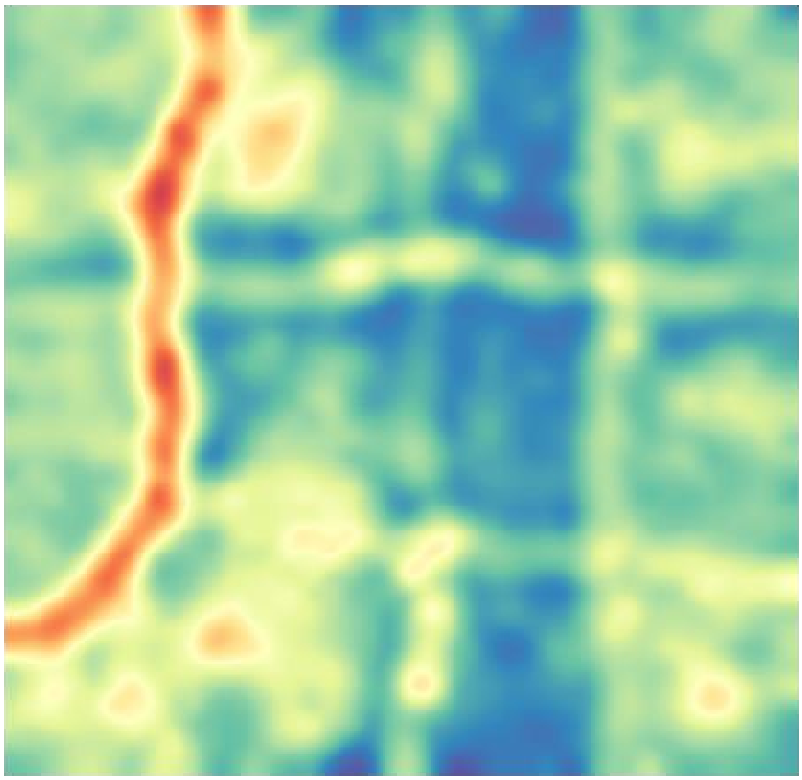

Рис. 2. SOM после этапа самоорганизации

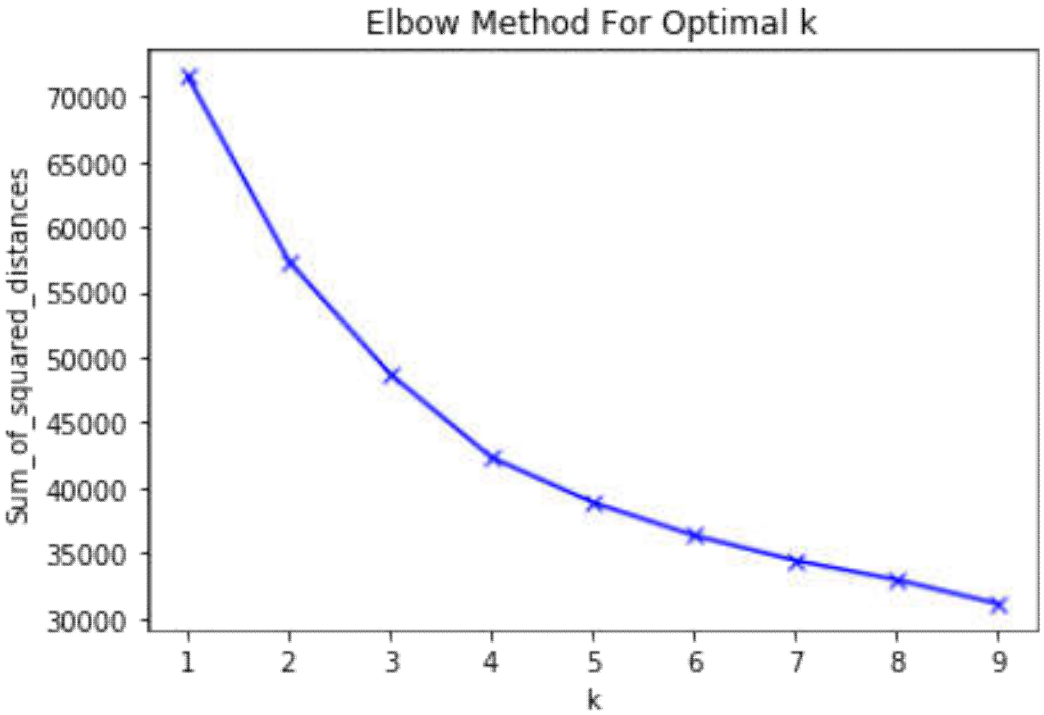

Рис. 3. Elbow method

Используем для кластеризации метод k-средних, который был импортирован из библиотеки Scikit-learn, указывая в параметрах найденное оптимальное количество кластеров и выводим получившийся результат на двумерной карте. (Рис.4)

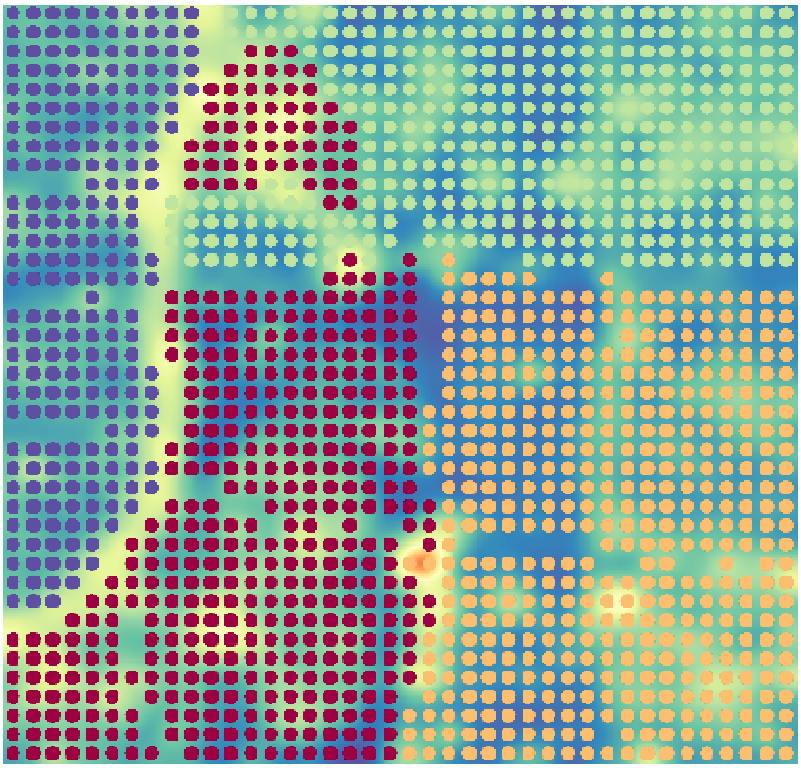

Рис. 4. SOM после этапа сходимости с 4 полученными кластерами

После кластеризации, мы можем выделить любую интересующую нас группу и повторить тот же алгоритм действий, т.е. определение количества нейронов согласно формуле (1), тренировку с двумя этапами, определение наиболее оптимального числа кластеров (Рис.5), а также кластеризацию и вывод полученных групп на карте (Рис. 6)

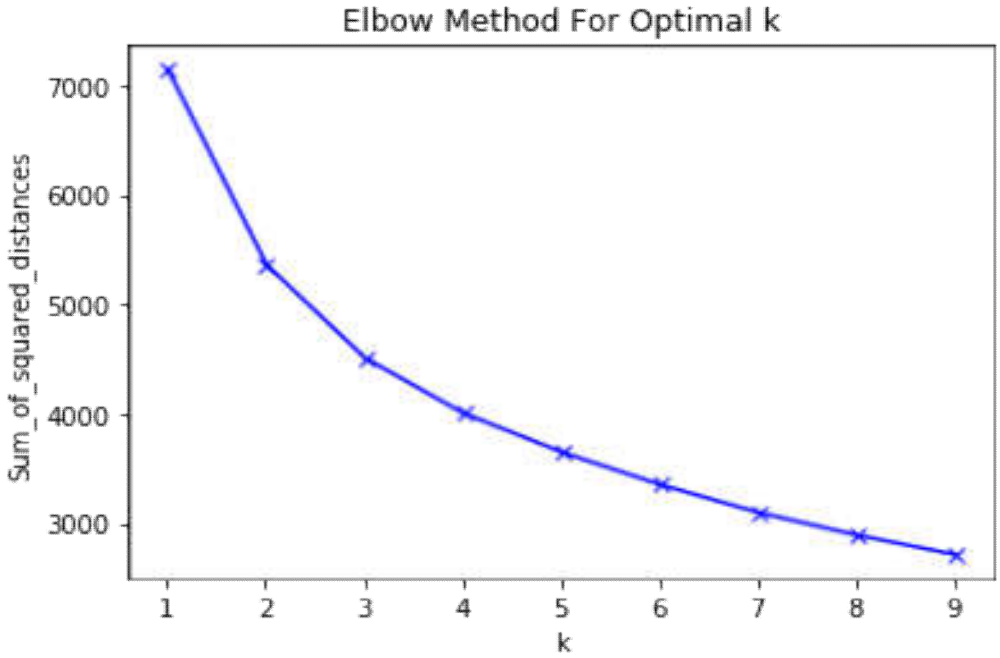

Рис. 5. Elbow method на примере данных нулевого кластера

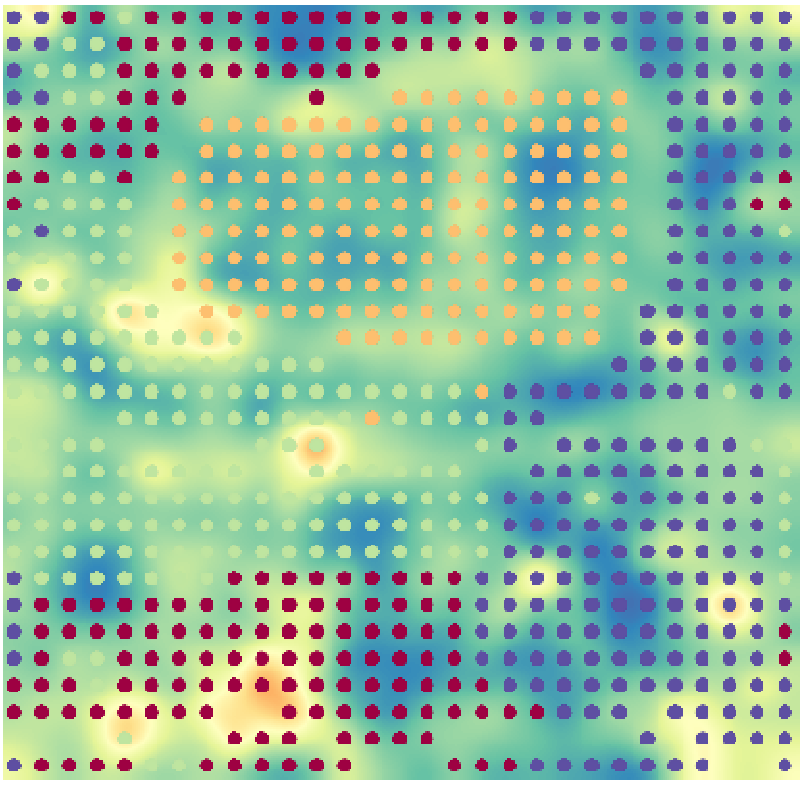

Рис. 6. SOM после этапа сходимости с 4 полученными кластерами на примере данных нулевого кластера

5 Заключение

В результате применения нейросетевого каскада, основанного на SOM с применением k-means в качестве метода кластеризации быстро и успешно могут решаться задачи представления и кластеризации большого объема данных.

Литература

1. Kaggle [Электронный ресурс] URL: https://www.kaggle.com/uciml/pm25-data-for-five-chinese-cities. (дата обращения 23.01.2019)
2. Github [Электронный ресурс] URL: https://github.com/peterwittek/somoclu (дата обращения 27.01.2019)
3. Jing Tian, Michael H. Azarian, and Michael Pecht. Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm. – Center for Advanced Life Cycle Engineering (CALCE), University of Maryland, College Park, MD, 20742, U.S.A. -9 c.
4. Scikit-learn [Электронный ресурс] URL: https://scikit-learn.org/0.20/ (дата обращения 03.02.2019)
5. Саймон Хайкин. Neural Networks: A Comprehensive Foundation, 2-е издание. М: Вильямс, 2018, - 1104 с

О.К. Головнин, А.В. Маркелов

НЕЙРОАССИСТЕНТ ДЛЯ СОСТАВЛЕНИЯ ИНДИВИДУАЛЬНОГО ПЛАНА ТРЕНИРОВОК

(Самарский университет)

Ведение здорового образа жизни неразрывно связано с систематическими физическими нагрузками и правильным питанием. Выбор эффективных и безопасных планов тренировок и питания может быть осуществлен при поддержке различных мобильных приложений, например [1-3]. Большинство мобильных приложений основываются на выборке из уже существующих библиотек планов тренировок, поэтому могут быть неэффективны и недостаточно гибки вследствие конечности возможных вариантов.

Развитие искусственных нейронных сетей и их интеграция с мобильными устройствами позволили сформироваться рынку нейроассистентов – мобильных приложений, оснащенных искусственным интеллектом и расширенными возможностями по взаимодействию с пользователем и анализу его действий и предпочтений. Таким образом, цель работы – разработка нейроассистента, главной функцией которого является составление индивидуального плана тренировок и питания.

Виртуальный нейроассистент за счет обученной нейросети подбирает оптимальные упражнения, которые будут интересны пользователю и