МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

САМООРГАНИЗУЮЩИЕСЯ КАРТЫ КОХОНЕНА И ИХ ПРИМЕНЕНИЕ В АНАЛИЗЕ ДАННЫХ

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

Студентки 2 курса 248 группы
Направления 09.04.03 — Прикладная информатика

механико-математического факультета
Бердниковой Алины Сергеевны

Научный руководитель
доцент, к.ф.м.н., доцент _____ М.Г.Плешаков

Заведующий кафедрой
д.ф.-м.н., доцент _____ С.П.Сидоров

Саратов 2020

ВВЕДЕНИЕ

В настоящее время в разных областях науки и деятельности человека (экономика, финансы, медицина, телекоммуникации, химия, биология, физика и т.п.) сформированы большие массивы разнородной информации. Такая информация может представлять собой переменные состояния каких-либо наблюдаемых объектов или процессов, например, машин, станков, комплексов и целых предприятий и храниться в виде электронных таблиц в базах данных.

В связи с этим актуальными становятся задачи анализа данных и получения в кратчайшие сроки информации о качественном распределении показателей, признаков и состояний изучаемых или используемых объектов на основании уже имеющейся о них информации с целью дальнейшего построения стратегии их применения и развития.

Специфика современных задач анализа данных такова, что часто для их решения предоставляется либо чрезмерно большой массив разнородных данных, либо, наоборот, количество данных для анализа мало и значения в некоторых их признаках отсутствуют или пропущены.

Для решения таких задач используются методы хранилищ данных, статистические методы, эволюционные алгоритмы, стохастические методы, методы нечеткой логики, методы искусственных нейронных сетей.

За последнее десятилетие машинное обучение беспрецедентно продвинулось в таких разных областях, как распознавание образов, робомобили и сложные игры, например, го. Эти успехи в основном были достигнуты через обучение глубоких нейронных сетей с одной из двух парадигм – обучение с учителем и обучение с подкреплением. Обе парадигмы требуют разработки человеком обучающих сигналов, передающихся затем компьютеру. В случае обучения с учителем – это «цели» (к примеру, правильная подпись под изображением); в случае с подкреплением это

«награды» за успешное поведение. Поэтому пределы обучения определяются людьми.

И если некоторые учёные считают, что достаточно обширной программы тренировок – к примеру, возможность успешно выполнить широкий набор задач – должно быть достаточно для порождения интеллекта общего назначения, то другие думают, что истинному интеллекту потребуются более независимые стратегии обучения.

Обучение без учителя – это парадигма, разработанная для создания автономного интеллекта путём награждения агентов (компьютерных программ) за изучение наблюдаемых ими данных безотносительно каких-то конкретных задач. Иначе говоря, агент обучается с целью обучиться.

Ключевая мотивация в обучении без учителя состоит в том, что если данные, передаваемые обучающимся алгоритмам имеют чрезвычайно богатую внутреннюю структуру (изображения, видеоролики, текст), то цели и награды в обучении обычно весьма сухие (метка «собака», относящаяся к этому виду, или единица/ноль, обозначающие успех или поражение в игре). Это говорит о том, что большая часть того, что изучает алгоритм, должна состоять из понимания самих данных, а не из применения этого понимания к решению определённых задач.

Тема работы является **актуальной** в силу широкого использования и развития средств кластеризации в анализе данных.

Целью данной работы является демонстрация использования самоорганизующихся карт Кохонена как одного из классических средств для проведения кластерного анализа. Для её достижения были сформулированы следующие **задачи**:

1. Изучить специфику структуры и функционирования карт Кохонена;
2. Рассмотреть программные средства реализации нейронных сетей;
3. Реализовать алгоритм самоорганизующихся карт Кохонена и применить его на практике.

Работа состоит из введения, 3 разделов, заключения, списка использованных источников и приложения. Общий объем работы составляет 52 страницы.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обосновывается актуальность темы работы, формулируется цель работы и решаемые задачи.

В **первом** разделе описывается структура и особенности сетей и карт Кохонена, рассматриваются алгоритмы обучения карт Кохонена.

Рассмотрим базовую структуру сети Кохонена (представлена на рисунке 1):

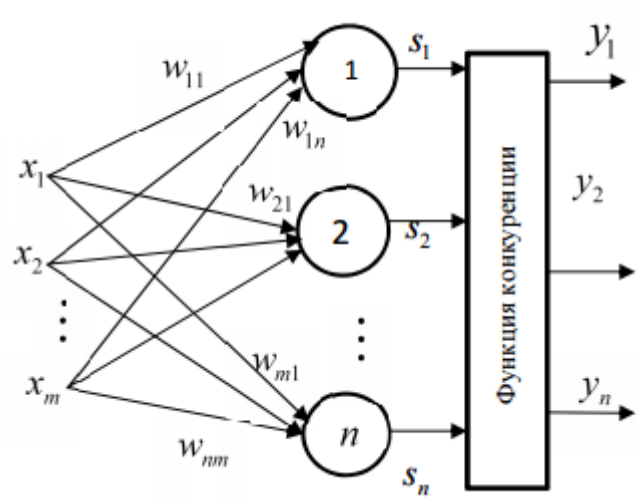

Рисунок 1 – Общая структура сети Кохонена

Слой Кохонена состоит из некоторого количества n параллельно действующих линейных элементов. Все они имеют одинаковое число входов m и получают на свои входы один и тот же вектор входных сигналов. Выход сумматора определяется как

$$s_j = w_{j0} + \sum_{i=1}^{m} w_{ij} x_i,$$

где j – номер нейрона, i – номер входа, w_{j0} – пороговый коэффициент, w_{ij} – вес i-го входа j-го нейрона.

Для обработки выходных сигналов слоя Кохонена используется функция конкуренции, для которой характерно правило «победитель забирает всё» (*WTA – Winner Takes All*), т.е. наибольший сигнал превращается в единичный, остальные обращаются в ноль. При этом, если максимум достигается на выходах нескольких сумматоров, то единичный выходной сигнал будет соответствовать только одному их них.

В сетях Кохонена используется обучение без учителя. При подаче на вход сети вектора x побеждает нейрон, для которого выполняется соотношение

$$d(x, w_j) = \min_{1 \le i \le n} d(x, w_i),$$

где $d(x, w)$ – расстояние между векторами x и w.

Номер нейрона-победителя определяет кластер, к которому должен был причислен входной вектор.

Для ускорения работы алгоритма рекомендуется нормировать входные значения. Для этого используется одна из формул:

$$x_{Ni} = \frac{x_i}{\sqrt{\sum_{i=1}^{m} x_i^2}}, x_{Ni} = \frac{x_i}{|x_i|},$$

где x_{Ni} – нормированный компонент входного вектора.

Сначала производится непосредственная инициализация сети – первоначальное задание векторов весов, в наиболее простом случае – случайное. Обучение сети Кохонена заключается в циклическом повторении шагов:

1. Подача исходных данных на входы.
2. Нахождение выхода каждого нейрона.
3. Определение нейрона-победителя.
4. Корректировка весов "выигравшего" нейрона по *правилу Кохонена*

$$w_i^{(k+1)} = w_i^{(k)} + \gamma_i^{(k)} \left[x - w_i^{(k)} \right],$$

где x – входной вектор,

k – номер цикла,

$\gamma_i^{(k)}$ – коэффициент скорости обучения.

5. Переход на шаг 1, если обучение не завершено.

Таким образом, нейрон, у которого вектор весов был ближе к входному вектору, обновляется, чтобы быть еще ближе. В результате он, скорее всего, выиграет конкуренцию при подаче на вход близкого вектора и проиграет при подаче существенно отличающегося вектора. После многократной подачи обучающих векторов будет иметься нейрон, который выдает 1, когда вектор принадлежит кластеру, и 0, когда вектор не принадлежит кластеру. Таким образом, сеть учится классифицировать входные векторы.

Самоорганизующиеся карты Кохонена (SOM) представляют из себя вычислительный метод, использующийся для задач кластеризации, а также для визуализации и анализа данных из пространств высокой размерности. В своем базовом варианте метод SOM создает граф подобия входных данных. Он преобразует нелинейные статистические соотношения между многомерными данными в простые геометрические связи между изображающими их точками на устройстве отображения низкой размерности, обычно в виде регулярной двумерной сетки узлов. Так как SOM осуществляет сжатие информации с сохранением в получаемом изображении наиболее значительных топологических и/или метрических связей между первичными элементами данных, можно также сказать, что она порождает некоторого вида обобщения. Два характерных свойства SOM – визуализацию и обобщение – можно использовать различными способами в решении сложных задач, таких как анализ процессов, машинное восприятие, управление, передача информации.

Целью применения данного метода является поиск неявных закономерностей в данных на основе снижения размерности исходного пространства в пространство меньшей размерности (на практике чаще всего

используется двумерное, по причине, в частности, удобной визуализации). При этом топология исходного пространства не меняется. В результате обучения данной модели получается решетка, состоящая из обученных нейронов. Она и называется "картой" исходного пространства.

В процессе обучения карты производится настройка весов не только победившего нейрона, но и его соседей. Таким образом, близкие по некоторой метрике входные векторы в сети Кохонена относятся к одному нейрону, который и является центром кластера. В случае карт Кохонена они могут относиться к близко расположенным на сетке, но разным нейронам.

На карте нейроны располагаются в узлах двумерной сетки с прямоугольными или шестиугольными ячейками (рисунок 2). Нейроны-соседи определяются расстоянием между нейронами на карте.

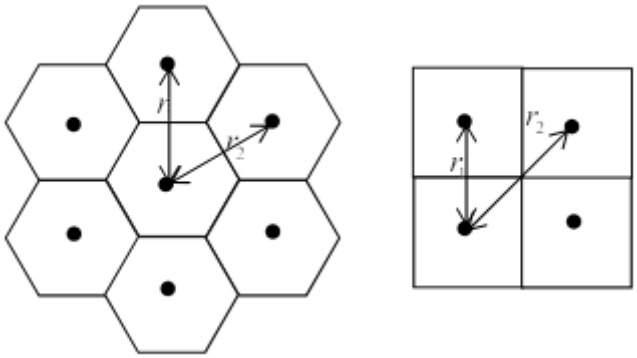

Рисунок 2 – Карты с прямоугольными и шестиугольными ячейками

Как видно из рисунка 2, шестиугольные ячейки отображают декартово расстояние корректнее, так как расстояния между центрами смежных ячеек будут одинаковыми.

Каждой ячейке соответствует нейрон сети Кохонена. То есть, в карте Кохонена число нейронов равно числу ячеек карты и больше числа нейронов сети Кохонена, равного числу кластеров.

Для каждого нейрона-ячейки вычисляется одна из статистических характеристик выбранного компонента входных векторов, попавших в ячейку.

В зависимости от величины этой характеристики ячейка окрашивается в тот или иной цвет.

Карты Кохонена позволяют по раскраске ячеек карты только выдвигать гипотезы о кластерной структуре, числу кластеров и зависимостях между значениями отдельных переменных. Выдвинутые гипотезы должны проверяться и подтверждаться иными способами.

Среди алгоритмов обучения наиболее часто используются три:

- Последовательный (Iterative SOM);
- Пакетный (Batch-Learning SOM);
- Алгоритм нейронного газа;

Пакетный алгоритм требует больших затрат по памяти, однако может сходиться в несколько раз быстрее последовательного. При этом наиболее быстрым является алгоритм нейронного газа.

Во **втором** разделе приводится описание различных программных средств реализации алгоритмов анализа данных. Особое внимание уделено библиотеке TensorFlow и среде MATLAB.

TensorFlow - это программная библиотека с открытым исходным кодом для численных расчетов с использованием графиков потоков данных. Первоначально она была разработанна исследователями и инженерами, работающими в команде Google Brain Team в исследовательской организации Google Machine Intelligence для целей машинного обучения и исследований глубоких нейронных сетей.

Google открыл основу своей системы машинного обучения, библиотеку TensorFlow, в конце 2015 года с разрешения компании Apache 2.0. До этого библиотека использовалась Google как патентованное средство распознавания речи, поиска, обработки фотографий и для электронной почты Gmail вместе с остальными приложениями.

Библиотека реализована на основе языка C++ и имеет удобный интерфейс прикладного программирования для Python. Благодаря простым зависимостям TensorFlow может быть быстро развернута в разных архитектурах.

Одним из лучших свойств TensorFlow является её способность автоматического дифференцирования, упрощающая выполнение обратного распространения ошибки обучения при использовании нейронных сетей.

Другой особенностью библиотеки является интерактивная среда визуализации под названием Tensorboard. Она может отслеживать ход выполнения программы, отображать итоговые журналы и демонстрировать блок-схему преобразования данных.

TensorFlow используется не только в нейронных сетях, но и при матричных вычислениях и манипуляциях данными, для которых она обладает готовыми пакетами, что делает её более универсальной по сравнению с другими библиотеками.

TensorFlow обладает хорошей документацией и официально поддерживается Google.

MATLAB – это высокоуровневый язык технических расчетов, интерактивная среда разработки алгоритмов и современный инструмент анализа данных. По сравнению с традиционными языками программирования MATLAB позволяет на порядок сократить время решения типовых задач и значительно упрощает разработку новых алгоритмов. MATLAB представляет собой основу всего семейства продуктов MathWorks и является главным инструментом для решения широкого спектра научных и прикладных задач, в таких областях как: моделирование объектов и разработка систем управления, проектирование коммуникационных систем, обработка сигналов и изображений, измерение сигналов и тестирование, финансовое моделирование, вычислительная биология и др.

Ядро MATLAB позволяет максимально просто работать с матрицами реальных, комплексных и аналитических типов данных и со структурами

данных и таблицами поиска. MATLAB содержит встроенные функции линейной алгебры (LAPACK, BLAS), быстрого преобразования Фурье (FFTW), функции для работы с полиномами, функции базовой статистики и численного решения дифференциальных уравнений; расширенные математические библиотеки для Intel MKL. Все встроенные функции ядра MATLAB разработаны и оптимизированы специалистами и работают быстрее или так же, как их эквивалент на C/C++.

Язык MATLAB является высокоуровневым языком программирования, включающим основанные на матрицах структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и интерфейсы к программам, написанным на других языках программирования. Программы, написанные на MATLAB, бывают двух типов — функции и скрипты. Функции имеют входные и выходные аргументы, а также собственное рабочее пространство для хранения промежуточных результатов вычислений и переменных. Скрипты же используют общее рабочее пространство. Как скрипты, так и функции не интерпретируются в машинный код и сохраняются в виде текстовых файлов. Существует также возможность сохранять так называемые pre-parsed программы — функции и скрипты, обработанные в вид, удобный для машинного исполнения. В общем случае такие программы выполняются быстрее обычных. Основной особенностью языка MATLAB является его широкие возможности по работе с матрицами. Математика и вычисления MATLAB предоставляет пользователю большое количество (несколько сотен) функций для анализа данных, покрывающие практически все области математики.

Одно из самых интересных особенностей среды MATLAB – возможность создавать специальные наборы инструментов (англ. toolbox), расширяющих его функциональность. Наборы инструментов представляют собой коллекции функций, напсанных на языке MATLAB для решения определённого класса задач. Компания Mathworks поставляет наборы инструментов, которые используются во многих областях, включая следующие: Цифровая обработка

сигналов, изображений и данных: DSP Toolbox, Image Processing Toolbox, Wavelet Toolbox, Communication Toolbox, Filter Design Toolbox — наборы функций, позволяющих решать широкий спектр задач.

Третий раздел посвящен реализации алгоритма самоорганизующихся карт Кохонена в среде MATLAB и с помощью библиотеки TensorFlow на языке Python.

Реализация в среде MATLAB показана на примере датасета «Ирисы Фишера» с помощью пакета Deep Learning Toolbox.

Реализация на языке Python включала в себя разработку класса SOM и применение этого класса при кластеризации датасета, содержащем информацию о пользовательских предпочтениях от 73516 пользователей в 12294 аниме с сайта myanimelist.net.

В заключении приведены результаты магистерской работы.

Основные результаты

Были изучены особенности самоорганизующихся карт Кохонена и реализованы в рассмотренных средах. Код программы частично представлен и в самой работе, и в **Приложении**. Основное отличие SOM от других моделей состоит в наглядности и удобстве использования. Эти сети позволяют упростить многомерную структуру, их можно считать одним из методов проецирования многомерного пространства в пространство с более низкой размерностью. SOM дает нам понимание структуры данных и позволяет решить задачу кластеризации.

Кластеризация пробного датасета выявила 3 кластера, не подлежащих точной интерпретации. Предполагается, что принадлежность пользователя к конкретному кластеру позволит выявить предпочитаемые жанры произведений, характерных для данного кластера. Данная информация потенциально может быть использована для систем рекомендации мультипликации.

Работа прошла апробацию на различных конференциях, в частности, на ежегодной студенческой конференции «Актуальные проблемы математики и механики», которую проводил механико-математический факультет СГУ в апреле 2019 года, в секции «Анализ данных», в VIII Международной молодежной научно-практической конференции «Математическое и компьютерное моделирование в экономике, страховании и управлении рисками», ноябрь 2019 года.