



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 Робототехника и комплексная автоматизация

КАФЕДРА Системы автоматизированного проектирования (РК-6)


РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

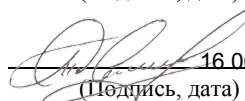
НА ТЕМУ

«Исследование итераций каскада и когерентных структур
методами машинного обучения»

Студент РК6-81Б
(Группа)

 16.06.2020 Д.М. Максимов
(Подпись, дата) (И.О. Фамилия)

Руководитель ВКР

 16.06.2020 А.П. Соколов
(Подпись, дата) (И.О. Фамилия)

Консультант

А.Ю. Першин
(Подпись, дата) (И.О. Фамилия)

Нормоконтролер

С.В. Грошев
(Подпись, дата) (И.О. Фамилия)

Москва, 2020 г.

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)

УТВЕРЖДАЮ

Заведующий кафедрой РК-6
(Индекс)

_____ А.П. Карпенко
(И.О.Фамилия)

«____» _____ 2020 г.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

Студент группы РК6-81Б

_____ Максимов Дмитрий Михайлович
(Фамилия, имя, отчество)

Тема выпускной квалификационной работы

Исследование итераций каскада и когерентных структур методами машинного обучения

Источник тематики (кафедра, предприятие, НИР): кафедра

Тема квалификационной работы утверждена распоряжением по факультету РК № 03.04.01-02/104 ВКР_Б от «25» октября 2019 г.

Часть 1. Аналитический обзор литературы

В рамках аналитического обзора литературы должны быть изучены основы теории динамических систем, включая способы их представления. Необходимо изучить модели и методы машинного обучения. Должна быть обоснована актуальность исследований.

Часть 2. Математическая постановка задачи.

В рамках раздела должна быть создана математическая модель дискретной и непрерывной динамической системы. Должны быть предложены методы моделирования и исследования динамических систем.

Часть 3. Проведение вычислительных экспериментов

Вычислительные эксперименты должны быть проведены с использованием разработанного в рамках ВКР программного обеспечения. Должна быть проведена верификация применяемых моделей на основе получаемых результатов вычислительных экспериментов.

Оформление курсовой работы:

Расчетно-пояснительная записка на 61 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.):

Работа содержит 41 рисунок.

5 листов-плакатов

Дата выдачи задания «18» февраля 2020 г.

Студент

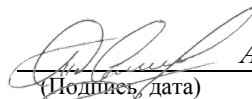


16.06.2020 Д.М. Максимов

(Подпись, дата)

(И.О.Фамилия)

Руководитель выпускной квалификационной работы



А.П. Соколов

(Подпись, дата)

(И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

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 РК

КАФЕДРА РК-6

ГРУППА РК6-81Б

УТВЕРЖДАЮ

Заведующий кафедрой РК-6
(Индекс)

А.П. Карпенко
(И.О.Фамилия)

«___» _____ 2020 г.

КАЛЕНДАРНЫЙ ПЛАН выполнения выпускной квалификационной работы

студента: Максимов Дмитрий Михайлович
(Фамилия, имя, отчество)

Тема выпускной квалификационной работы: исследование итераций каскада и когерентных структур методами машинного обучения

№ п/ п	Наименование этапов выпускной квалификационной работы	Сроки выполнения		Отметка о выполнении	
		план	факт	Должность	ФИО, подпись
1.	Задание на выполнение работы. Формулирование проблемы, цели и задач работы	<u>18.02.2020</u> Планируемая дата	<u>18.02.2020</u>	Руководитель ВКР	А.П. Соколов
2.	1 часть: <u>аналитический обзор литературы</u>	<u>18.02.2020</u> Планируемая дата	<u>31.03.2020</u>	Руководитель ВКР	А.П. Соколов
3.	Утверждение окончательных формулировок решаемой проблемы, цели работы и перечня задач	<u>28.02.2020</u> Планируемая дата	<u>28.02.2020</u>	Заведующий кафедрой	А.П. Карпенко
4.	2 часть: <u>математическая постановка задачи, разработка архитектуры программной реализации. программная</u>	<u>31.03.2020</u> Планируемая дата	<u>31.03.2020</u>	Руководитель ВКР	А.П. Соколов
5.	3 часть: <u>проведение вычислительных экспериментов, отладка и тестирование</u>	<u>30.04.2020</u> Планируемая дата	<u>30.04.2020</u>	Руководитель ВКР	А.П. Соколов
6.	1-я редакция работы	<u>31.05.2020</u> Планируемая дата	<u>31.05.2020</u>	Руководитель ВКР	А.П. Соколов
7.	Подготовка доклада и презентации	<u>17.06.2020</u> Планируемая дата	<u>17.06.2020</u>		
8.	Заключение руководителя	<u>15.06.2020</u> Планируемая дата	<u>15.06.2020</u>	Руководитель ВКР	А.П. Соколов
9.	Допуск работы к защите на ГЭК (нормоконтроль)	<u>15.06.2020</u> Планируемая дата	<u>11.06.2020</u>	Нормоконтролер	С.В. Грошев
10.	Внешняя рецензия	<u>12.06.2020</u> Планируемая дата	<u>12.06.2020</u>		
11.	Защита работы на ГЭК	<u>17.06.2020</u> Планируемая дата	<u>17.06.2020</u>		

Студент Максимов Д.М. Максимов
(подпись, дата)

Руководитель работы А.П. Соколов
(подпись, дата)

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)

**НАПРАВЛЕНИЕ
НА ГОСУДАРСТВЕННУЮ ИТОГОВУЮ АТТЕСТАЦИЮ**

Председателю
Государственной Экзаменационной Комиссии № _____

факультета «Робототехники и комплексной автоматизации» МГТУ им. Н.Э. Баумана

Направляется студента Максимов Дмитрий Михайлович группы РК6-81Б

на защиту выпускной квалификационной работы Исследование итераций каскада и когерентных структур методами машинного обучения

Декан факультета _____ «____» _____ 2020 г.

Справка об успеваемости

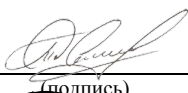
Студент Максимов Дмитрий Михайлович за время пребывания в МГТУ имени Н.Э. Баумана с 2016 г. по 2020 г. полностью выполнил учебный план со следующими оценками: отлично – _____ %, хорошо – _____ %, удовлетворительно – _____ %.

Инспектор деканата _____

Отзыв руководителя выпускной квалификационной работы

Студент Максимов Д.М. в процессе выполнения ВКР проявил себя как настоящий исследователь: самостоятельно изучил область знаний поставленной задачи, провёл анализ возможных вычислительных методов исследования, создал программные реализации анализируемых методов, провёл вычислительные эксперименты. В результате полученных численных результатов обосновал сделанные выводы. Работа выполнена автором самостоятельно и в полном объёме в полном соответствии с заданием и календарным планом. Максимов Д.М. достоин «отличной» оценки и присвоения квалификации бакалавр техники и технологий по направлению «Информатика и вычислительная техника».

Руководитель ВКР


(подпись)

А.П. Соколов

(ФИО)

« 16 » _____ июня 2020 г.

(дата)

Студент


(подпись)

Д.М. Максимов

(ФИО)

« 16 » _____ июня 2020 г.

(дата)

РЕФЕРАТ

Работа посвящена анализу динамических систем с помощью машинного обучения. Целью работы является построение итераций каскада для дискретных динамических систем и нахождение когерентных структур в турбулентных потоках. Каскад моделируется при помощи глубокой нейронной сети. Выделение когерентных структур сводится к задаче кластеризации. Рассматриваемый подход позволяет исследовать динамические системы как объект в задачах машинного обучения.

Тип работы: выпускная квалификационная работа.

Тема работы: анализ динамических систем с помощью машинного обучения.

Объект исследований: динамические системы.

СОДЕРЖАНИЕ

РЕФЕРАТ	6
СОДЕРЖАНИЕ	7
ВВЕДЕНИЕ.....	8
1. ОБЗОР ЛИТЕРАТУРЫ.....	9
1.1. ДИНАМИЧЕСКИЕ СИСТЕМЫ	9
1.2. МАШИННОЕ ОБУЧЕНИЕ	10
2. СОВРЕМЕННЫЕ РАЗРАБОТКИ.....	12
4. ВВЕДЕНИЕ В ДИНАМИЧЕСКИЕ СИСТЕМЫ	13
5. МОДЕЛИРОВАНИЕ ДИСКРЕТНЫХ ДИНАМИЧЕСКИХ СИСТЕМ.....	15
5.1. ВВЕДЕНИЕ В ДИСКРЕТНЫЕ ДИНАМИЧЕСКИЕ СИСТЕМЫ	15
5.2. ВВЕДЕНИЕ В НЕЙРОННЫЕ СЕТИ.....	15
5.3. МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ МОДЕЛИРОВАНИЯ КАСКАДОВ.....	18
5.3.1. ФОРМАЛЬНОЕ ПРЕДСТАВЛЕНИЕ КАСКАДОВ.....	18
5.3.2. ВВЕДЕНИЕ В РЕКУРРЕНТНЫЕ НЕЙРОННЫЕ СЕТИ (РНС)	19
5.3.3. ПОСТАНОВКА ЗАДАЧИ.....	21
5.3.4. ПОСТРОЕНИЕ РНС	22
5.3.5. МЕТОД МАКСИМИЗАЦИИ ПРАВДОПОДОБИЯ.....	23
5.3.6. ОПТИМИЗАЦИЯ ПАРАМЕТРОВ НЕЙРОННОЙ СЕТИ.....	24
5.4. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ И АНАЛИЗ РЕЗУЛЬТАТОВ.....	27
6. НАХОЖДЕНИЕ КОГЕРЕНТНЫХ СТРУКТУР В ТУРБУЛЕНТНЫХ ПОТОКАХ	28
6.1. ВВЕДЕНИЕ В ТУРБУЛЕНТНЫЕ ПОТОКИ.....	28
6.2. МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКОЙ СИСТЕМЫ КАК ТУРБУЛЕНТНОГО ПОТОКА.....	29
6.2.1. УРАВНЕНИЕ КУРАМОТО-СИВАШИНСКОГО.....	30
6.3. ИССЛЕДОВАНИЕ КОГЕРЕНТНЫХ СТРУКТУР С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ	32
6.3.1. ВВЕДЕНИЕ В ЗАДАЧУ	32
6.3.2. ВВЕДЕНИЕ В КЛАСТЕРИЗАЦИЮ	32
6.3.3. МЕТОД K-СРЕДНИХ.....	34
6.3.4. EXHRESTATION MAXIMIZATION (ЕМ-АЛГОРИТМ).....	35
6.3.5. АГЛОМЕРАТИВНАЯ ИЕРАРХИЧЕСКАЯ КЛАСТЕРИЗАЦИЯ	37
6.3.6. МЕТОД ОСНОВАННЫЙ НА ПЛОТНОСТИ	39
6.3.7. ИСПОЛЬЗОВАНИЕ МЕТОДОВ КЛАСТЕРИЗАЦИИ ДЛЯ ИССЛЕДОВАНИЯ КОГЕРЕНТНЫХ СТРУКТУР.....	41
6.4. АНАЛИЗ РЕЗУЛЬТАТОВ.....	51
ЗАКЛЮЧЕНИЕ	52
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	53
ПРИЛОЖЕНИЕ А	56

ВВЕДЕНИЕ

Динамические системы активно применяются как в фундаментальной, так и в прикладной математике, которая включает модели и методы во многих областях: физики, астрономии, биологии, метрологии, экономики и т.д. В прикладной математике в качестве динамической системы могут служить коробка с молекулами газа в физике, популяция видов в биологии, финансовый рынок в экономике или ветровые течения в метрологии. В фундаментальной математике динамическая система обычно представляется в виде системы дифференциальных уравнений.

По временной характеристике динамические системы разделяют на дискретные или каскады и на непрерывные или потоки. Например, можно ежегодно фиксировать количество людей населения и анализировать рост популяции из года в года, что будет примером дискретной динамической системы. В качестве непрерывной динамической системы можно представить линейный поток на торе.

Современные методы в области машинного обучения позволяют решить многие задачи, связанные с динамическими системами. С помощью таких моделей машинного обучения, как рекуррентные нейронные сети, возможно создание математической модели каскадов, а также построение необходимого количества итераций каскада.

Отсутствие понимания того, как искусственный интеллект достигает результатов, является одной из причин низкого уровня доверия к современным технологиям искусственного интеллекта. Именно поэтому объяснение работы методов машинного обучения является одним из важнейших направлений в российской национальной стратегии развития искусственного интеллекта [1].

1. ОБЗОР ЛИТЕРАТУРЫ

1.1. ДИНАМИЧЕСКИЕ СИСТЕМЫ

Общее представление динамической системы подробно описано в статье [2]. В статье динамическая система рассматривается как модель для описания и прогнозирования взаимодействия во времени между несколькими компонентами явления, которые рассматриваются как система. В статье указываются следующие компоненты динамической системы.

- Динамический компонент указывает, что время является неотъемлемым элементом модели. В динамических моделях время имеет основополагающее значение как для базовой структуры данных, так и для понимания того, как разворачивается процесс.
- Системный компонент предполагает, что исследовательские вопросы позиционируются с участием нескольких взаимодействующих компонентов большего целого. В контексте динамической системы это означает, что взаимодействующие компоненты ведут себя упорядоченно, следуя правилам, которые могут быть идентифицированы и определены.
- Модельный компонент указывает, что динамические связи между компонентами системы представлены в виде формальных математических уравнений.

В некоторых моделях динамических систем данные организованы по времени как последовательность повторных наблюдений данной переменной во времени, называемая данными временных рядов. В статье [3] рассматриваются модели временных рядов для представления развития системы. Статьи [4], [5] описывают динамические модели, где распределения времени отклика учитываются при формулировании и прогнозировании моделей (например, одновременное моделирование вероятности выбора и времени отклика выбора при прогнозировании выбора).

Описание основных элементов модели динамической системы подробно

представлено в статье [6]:

- состояние системы, которое представляет всю системную информацию в определенный момент времени;
- пространство состояний системы, которое представляет все возможные состояния системы, которые могут возникнуть;
- функция перехода состояния, которая описывает, как состояния системы изменяется со временем.

Значение и применение дискретных динамических систем описано в работе [7]. В данном источнике каскад – это динамическая система с дискретным временем, где временная переменная моделируется как дискретная, а временная задержка встроена в систему.

В работе [8] рассматриваются неавтономные динамические системы с дискретным временем. В центре внимания данной работы две формулировки дискретных по времени неавтономных динамических систем:

- двухпараметрические полугруппы;
- системы с косыми произведениями;

Основные концепции теории непрерывных динамических систем описано в источнике [9].

1.2. МАШИННОЕ ОБУЧЕНИЕ

В книге [10] описаны основы статистической теории машинного обучения, игр с предсказаниями и прогнозирования с применением экспертной стратегии. Также рассмотрены следующие алгоритмы:

- алгоритм взвешенного большинства;
- вероятностный алгоритм;
- алгоритм экспоненциального взвешивания;
- агрегирующий алгоритм Вовка;

Основные возможности применения и перспективы развития нейронных сетей описаны в статье [11]. Из основных направлений можно выделить:

- поиск информации;

- распознавание изображений;
- перевод;
- воспроизведение речи;

В статье [12] дана классификация нейронных сетей по структуре, количеству слоев, типу связей, структуре нейрона и т.д.

Проблема тренировки нейронной сети и алгоритм обратного распространения ошибки описаны в работе [13]. Основной проблемой тренировки является переобучение. Эта проблема возникает при долгом обучении сети на одних и тех же данных.

В книге [14] раскрыты основные математические принципы, лежащие в основе нейронных сетей и разобран пример нейросети, распознающей написанные от руки цифры.

Источник [15] иллюстрирует графические модели для описания распределения вероятностей, использующих Байесовский метод для распознавания образов и алгоритмы приближенного вывода ситуаций, в которых точные ответы получить невозможно.

Процесс тренировки генеративной сверточной нейронной сети для генерации изображений объектов по типу и цвету с интерполяцией рядов изображений и заполнением «пустых мест» недостающими элементами разобран в работе [16].

В статье [17] представлены результаты аналитического исследования рекуррентных нейронных сетей (РНС) и их обобщающая классификация, выполненная с позиций динамических систем. В работе выделены основные динамические режимы работы РНС, а также определены наиболее перспективные направления в развитии методов обучения РНС с учетом выявленных достоинств и недостатков существующих подходов.

В источнике [18] описаны следующие виды РНС:

- Long Short-Term Memory (LSTM) – долгая краткосрочная память;
- Gated Recurrent Unit (GRU).

Также выделены преимущества каждой нейросети в соответствии с поставленной задачей.

В статье [19] разобраны способы решения проблемы в облачном центре обработки данных с помощью прогнозирования рабочей нагрузки. Модель прогнозирования рабочей нагрузки разработана с использованием сетей с кратковременной памятью (LSTM). Предложенная модель протестирована на трех эталонных наборах журналов веб-сервера.

Применение рекурсивных рекуррентных нейронных сетей для моделирования процесса декодирования и синтаксического разбора в статистическом машинном переводе предложено в работе [20].

Общие принципы методов обучения нейронных сетей приведены в статье [21].

В материалах [22], [23] подробно описаны основные концепции и методы в машинном обучении, в частности нейронных сетях.

Источники [24], [25] служат в качестве справочного для практической реализации методов машинного обучения.

2. СОВРЕМЕННЫЕ РАЗРАБОТКИ

Человечество всегда стремилось познать сущность человека. Развитие компьютерных технологий привело к разработке систем, проявляющих поведение, свойственное человеку, такие системы называются системами искусственного интеллекта (ИИ).

Для современного этапа развития ИИ характерно продвижение разработок искусственных нейронных сетей (ИНС), такие сети имитируют работу биологических нейронов живых существ.

Прорыв в обучении машин произошел в 2016г., когда нейронная сеть AlphaGO обыграла в игру «го» сильнейшего людского представителя в данной игре. Данная сеть разработана с использованием технологии «глубинного машинного обучения», которая сейчас широко развивается. Развитием технологии глубинного обучения стала реализованная компанией IBM в 2017г.

технология распределенного глубинного обучения, позволяющая сократить время обучения нейронной сети [26].

3. ПОСТАНОВКА ЗАДАЧИ

Объект исследования: динамические системы.

Ключевые слова: «Динамические системы» («Dynamical systems»), «Непрерывные динамические системы» («Continuous-time dynamical systems»), «Дискретные динамические системы» («Discrete dynamical systems»), «Каскады» («Maps»), «Потоки» («Flows»), «Когерентные структуры» («Coherent structures»), «Машинное обучение» («Machine learning»), «Нейронные сети» («Neural networks»), «Уравнение Курамото-Сивашинского» («Kuramoto-Sivashinsky equation»).

Цель работы: исследовать возможность применения моделей и методов машинного обучения для анализа динамических систем.

Поставленные и реализованные **задачи**.

- протестировать нейронную сеть для моделирования дискретной динамической системы;
- предложить методы машинного обучения для выявления когерентных структур в непрерывных динамических системах;

Требуется получить:

- нейронную сеть для предсказаний итераций каскада;
- методы машинного обучения для обнаружения когерентных структур на примере уравнения Курамото-Сивашинского.

4. ВВЕДЕНИЕ В ДИНАМИЧЕСКИЕ СИСТЕМЫ

Динамические системы - это раздел математики, посвященный изучению систем, управляемых согласованным набором законов во времени, таких как разностные и дифференциальные уравнения. Акцент динамических систем заключается в понимании геометрических свойств траекторий и долговременного поведения. За последние 40 лет, с открытием хаоса и странных

аттракторов, теория динамических систем приобрела значительный интерес, и было обнаружено, что она имеет связи со многими различными областями математики (такими как теория чисел и топология) и науки. Динамические системы могут моделировать невероятный диапазон поведения, такой как движение планет в солнечных системах, распространение болезней среди населения, форма и рост растений, взаимодействие оптических импульсов или процессы, которые регулируют электронные цепи и сердцебиение.

Для создания динамической системы необходимо выбрать что будет развиваться со временем и какое правило определяет, как будет проходить развитие со временем. Таким образом, динамическая система - это модель, описывающая временную эволюцию системы.

Первым шагом в создании динамической системы является определение того, что является основой, которое будет развиваться со временем. Чтобы сделать это, нужно создать набор переменных, которые дают полное описание системы в любой конкретный момент времени.

Под «полным описанием» не обязательно подразумевается, что переменные будут полностью описывать реальную моделируемую систему. Но переменные должны полностью описывать состояние математической системы. В динамической системе, если значения этих переменных в определенное время известны, то известно все о состоянии системы в это время.

Переменные, которые полностью описывают состояние динамической системы, называются переменными состояния. Множество всех возможных значений переменных состояния является пространством состояний.

Пространство состояний может быть дискретным и состоять из изолированных точек, например, если переменные состояния могут принимать только целые значения. Он может быть непрерывным, состоящим из гладкого набора точек, например, если переменные состояния могут принимать любое действительное значение. В случае, когда пространство состояний является непрерывным и конечномерным, его часто называют фазовым пространством, а число переменных состояния является размерностью динамической системы.

Пространство состояний также может быть бесконечномерным.

Вторым шагом в создании динамической системы является определение правила эволюции динамической системы во времени. Это правило должно быть определено, чтобы переменные состояния были полным описанием состояния системы в следующем смысле: значение переменных состояния в конкретный момент времени должно полностью определять эволюцию всех будущих состояний. Если эволюция во времени зависит от переменной, не включенной в пространство состояний, то правило, объединенное с пространством состояний, не определяет динамическую систему. Нужно либо изменить правило, либо дополнить пространство состояний необходимыми переменными, чтобы сформировать динамическую систему.

5. МОДЕЛИРОВАНИЕ ДИСКРЕТНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

5.1.ВВЕДЕНИЕ В ДИСКРЕТНЫЕ ДИНАМИЧЕСКИЕ СИСТЕМЫ

Динамические системы, в которых состояние системы развивается с дискретными временными шагами, называются дискретными динамическими системами или каскадами.

При моделировании системы как каскада, делается состояние системы в определенной последовательности. Состояния могут появляться один раз в год, раз в миллисекунду или даже нерегулярно.

Идея заключается в том, что записывается любая переменная, определяющая состояние системы: выбранные переменные состояния, которые эволюционируют через пространство состояний. Чтобы завершить описание динамической системы, нужно указать правило, которое определяет, с учетом начального состояния, какой должна быть результирующая последовательность будущих состояний.

5.2.ВВЕДЕНИЕ В НЕЙРОННЫЕ СЕТИ

Нейронная сеть — это последовательность нейронов, соединенных между собой синапсами. Структура нейронной сети пришла в мир программирования

прямоком из биологии. Благодаря такой структуре, машина обретает способность анализировать и даже запоминать различную информацию. Нейронные сети также способны не только анализировать входящую информацию, но и воспроизводить ее из своей памяти. Другими словами, нейросеть это машинная интерпретация мозга человека, в котором находятся миллионы нейронов, передающих информацию в виде электрических импульсов.

Нейронная сеть считается универсальной моделью, так как она способна аппроксимировать любые поверхности. Соответствующую теорему сформулировал в 1957 году Андрей Николаевич Колмагоров.

Теорема (А.Н. Колмагоров, 1957) Каждая непрерывная функция $a(x)$, заданная на единичном кубе d -мерного пространства, представима в виде

$$a(x) = \sum_{i=1}^{2d+1} \sigma_i \left(\sum_{j=1}^d f_{ij}(x_j) \right), \quad (5.2.1)$$

где $x = [x_1, \dots, x_d]^T$ – вектор описания объекта, функции $\sigma_i(\cdot)$ и $f_{ij}(\cdot)$ являются непрерывными, а f_{ij} не зависят от выбора a .

Согласно формулировке теоремы, функция $a(x)$ определена только на единичном кубе, следовательно все элементы выборки должны лежать в нем. Это не является проблемой: всегда можно отмасштабировать признаки так, чтобы на обучающей выборке каждый признак принимал значения из отрезка $[0, 1]$.

Также следует отметить, что в теореме не указан конкретный вид функции σ_i и f_{ij} . Проблема конструирования нейронной сети остается сложной задачей до сих пор.

Нейрон — это вычислительная единица, которая получает информацию, производит над ней простые вычисления и передает ее дальше. Однослойная нейронная сеть, или нейрон, определяется выражением:

$$a(x, \omega) = \sigma(\omega^T x) = \sigma \left(\sum_{j=1}^d \omega_j^{(1)} x_j + \omega_0^{(1)} \right), \quad (5.2.2)$$

где σ – функция активации, ω – вектор параметров (весов), x – вектор описания

объекта. Функция активации должна быть непрерывной, монотонной и, желательно, дифференцируемой функцией. Следует обратить внимание, что для удобства $x \in \mathbb{R}^{d+1}$ дополнен постоянным на всех объектах признаком $x_0 = 1$. Соответствующий ему вклад в скалярное произведение $\omega^T x = \omega_0$.

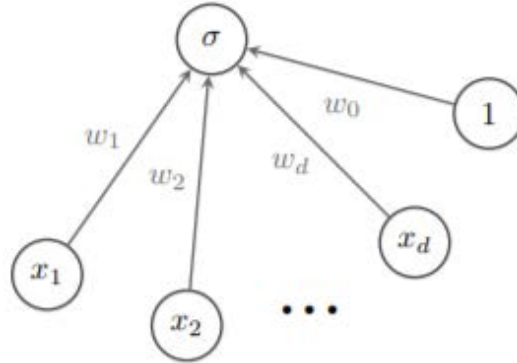


Рис. 5.2.1: нейрон можно изобразить в виде вершины графа: он характеризуется своей функцией активации, имеет один выход и множество входов

Если нейросеть состоит из большого количества нейронов вводят термин слоя. Слои делятся на:

- входной (получает информацию);
- скрытый (обрабатывает информацию);
- выходной (выводит результат).

Нейроны соединяются синапсами. У синапсов есть 1 параметр – вес. Общий вид нейронной сети указан на рис. 5.2.2.

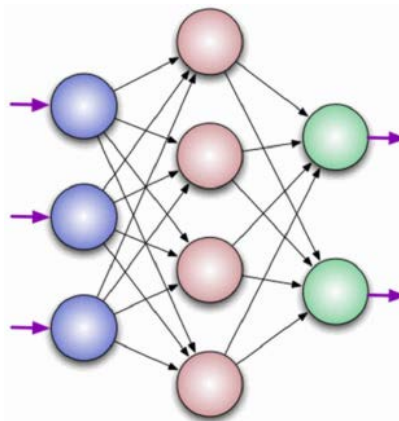


Рис. 5.2.2 Модель нейронной сети, где синим цветом указаны входные нейроны, скрытые – красным, выходные – зеленым. Черные стрелки – синапсы, фиолетовые – входные и выходные данные

5.3. МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ МОДЕЛИРОВАНИЯ КАСКАДОВ

5.3.1. ФОРМАЛЬНОЕ ПРЕДСТАВЛЕНИЕ КАСКАДОВ

В соответствии с пунктом 5.1 классическая форма каскада:

$$S^{(t)} = f(S^{(t-1)}; \theta), \quad (5.3.1.1)$$

где $S^{(t)}$ – состояние системы в момент времени t , θ – параметр динамической системы.

Выражение (5.3.1.1) рекуррентное, так как состояние S в момент времени t ссылается на состояние в момент времени $t - 1$.

Пусть τ – конечное число временных шагов каскада, тогда каскад можно представить в виде графа, примени определение (5.3.1.1) $\tau - 1$ раз. Так если развернуть выражение (5.3.1.1) для $\tau = 3$ шагов:

$$S^{(3)} = f(S^{(2)}; \theta) = f(f(S^{(1)}; \theta); \theta), \quad (5.3.1.2)$$

Так как состояние $S^{(0)}$ по определению динамической системы всегда задано, то путем повторного применения определения (5.3.1.1) получится выражение, не содержащее рекурсии. Такое выражение можно представить традиционным ациклическим ориентированным графом вычислений.

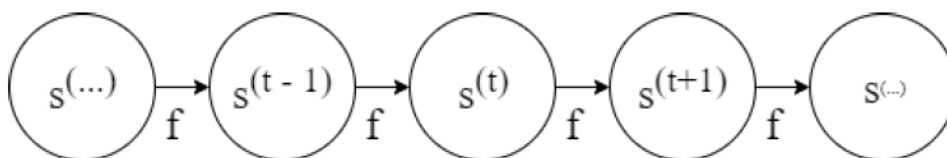


Рис. 5.3.1.1 Классическая дискретная динамическая система, описываемая выражением (5.3.1.1), в виде развернутого графа вычислений. Каждая вершина представляет состояние в некоторый момент t , а функция f отображает состояние в момент t на состояние в момент $t+1$. Одни и те же параметры (значение θ , параметризующее f) используется на всех временных шагах.

Дискретная динамическая система, управляемая внешним сигналом $x^{(t)}$ имеет вид:

$$S^{(t)} = f(S^{(t-1)}, x^{(t)}; \theta), \quad (5.3.1.3)$$

Теперь состояние содержит всю информацию обо всей прошлой последовательности.

Проанализировав виды нейронных сетей и выражение (5.3.1.3), можно сделать вывод о необходимости использования рекуррентной нейронной сети для моделирования каскада.

5.3.2. ВВЕДЕНИЕ В РЕКУРРЕНТНЫЕ НЕЙРОННЫЕ СЕТИ (РНС)

Чтобы подчеркнуть, что состояние динамической системы – это на самом деле скрытые блоки сети, следует переписать уравнение (5.3.1.3), используя для представления состояния переменную h :

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta). \quad (5.3.2.1)$$

РНС, удовлетворяющая выражению (5.3.2.1), показана на рис. 3.

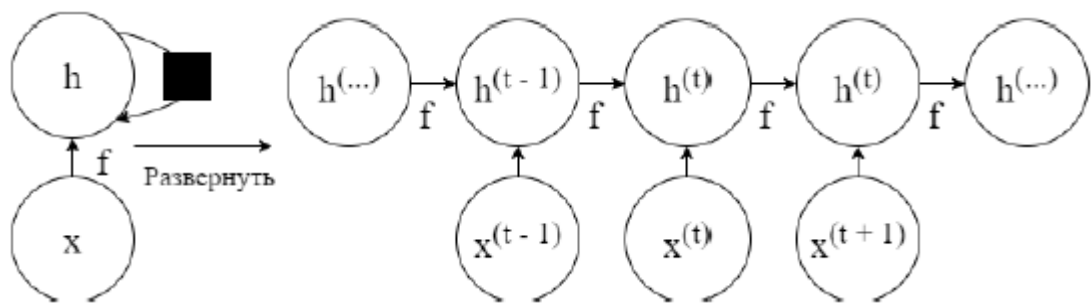


Рис. 5.3.2.1 Рекуррентная сеть без выходов. Эта сеть обрабатывает информацию из входа x , включая ее в состояние h , которое передается дальше во времени. (Слева) Принципиальная схема. Черный квадрат обозначает задержку на один временно шаг. (Справа) Та же сеть в виде развернутого графа вычислений, в котором каждая вершина ассоциирована с одним моментом времени

Когда РНС обучают решать задачу, в которой требуется предсказывать будущее по прошлому, сеть обычно обучается использовать $h^{(t)}$ как сводку относящихся к задаче аспектов последовательности входных данных, предшествующей моменту времени t . В общем случае в сводке по необходимости утрачивается часть информации, потому что она отображает последовательность произвольной длины $(x^{(t)}, x^{(t-1)}, \dots, x^{(1)})$ на вектора фиксированной длины $h^{(t)}$. В зависимости от критерия обучения некоторые аспекты прошлой последовательности могут запоминаться в сводке с большой

точностью, чем остальные.

Можно представить развернутое рекуррентное выражение после t шагов функции $h^{(t)} = g^{(t)}(x^{(t)}, x^{(t-1)}, \dots, x^{(1)}) = f(h^{(t-1)}, x^{(t)}; \theta)$.

Функция $g^{(t)}$ принимает на входе всю прошлую последовательность $(x^{(t)}, x^{(t-1)}, \dots, x^{(1)})$ и порождает текущее состояние, но развернутая рекуррентная структура позволяет представить $g^{(t)}$ в виде многократного применения функции f . Таким образом процесс развертки дает два важных преимущества.

- Независимо от длины последовательности размер входа обученной модели всегда один и тот же, поскольку, он описывается в терминах перехода из одного состояния в другое, а не в терминах истории состояний переменной длины.
- Одну и ту же функцию перехода f с одними и теми же параметрами можно использовать на каждом шаге

Эти два фактора позволяют обучить одну модель f , которая действует на всех временных шагах и для последовательностей любой длины, не прибегая к обучению отдельных моделей $g^{(t)}$ для каждого временного шага.

РНС можно классифицировать по типу связи между входными и выходными данными (Рис. 5.3.2.2 – 5.3.2.5).



Рис. 5.3.2.2 РНН с типом связи один к одному (по сути является обычной нейронной сетью)

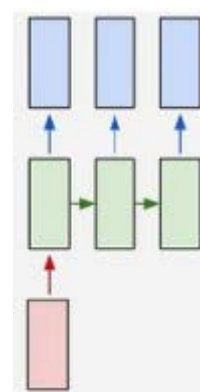


Рис. 5.3.2.3 РНН с типом связи один ко многим (один вход ко многим выходам)

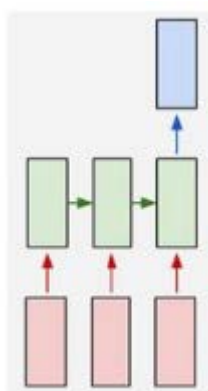


Рис. 5.3.2.4 РНН с типом связи многие к одному (много входов к одному выходу)

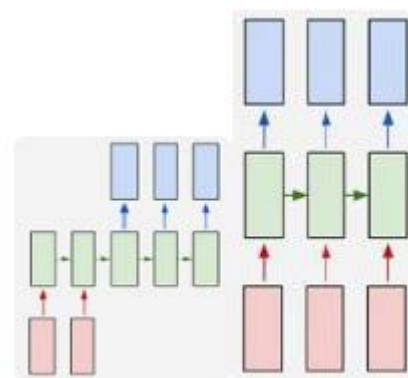


Рис. 5.3.2.5 РНН с типом связи многие ко многим (много входов ко многим выходам)

5.3.3. ПОСТАНОВКА ЗАДАЧИ

Из разнообразия РНС можно сделать вывод, что разные РНС решают разные задачи. Чтобы выбрать вид РНС необходимо установить условие задачи.

Пусть в качестве каскада будет видеоряд, тогда задача будет звучать так: по n кадрам построить видео из n кадров, где выходной кадр момента времени t должен совпадать со входным кадром момента времени $t + 1$, а выходной кадр для последнего временного состояния, должен не нарушать общую тенденцию каскада.

Поставленная задача в виде развернутого графа имеет вид:

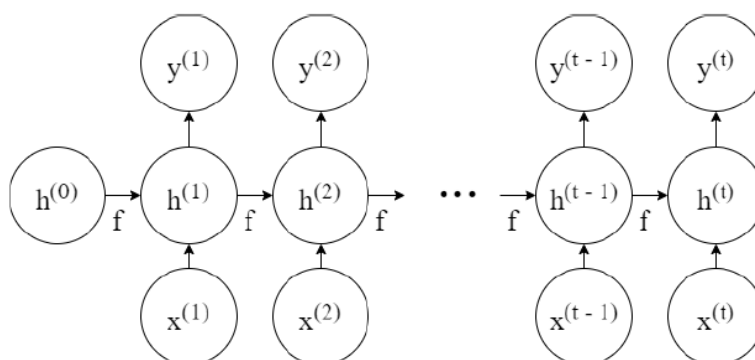


Рис. 5.3.2.5 Развернутый граф в соответствии с поставленной задачей,

где $x^{(i)}$ - i -ый входной кадр, а $y^{(i)}$ - i -ый выходной кадр,

$$i \in [1, \dots, t]$$

Данный граф имеет много входов и много выходов, что соответствует РНС

с типом связи многие ко многим.

5.3.4. ПОСТРОЕНИЕ РНС

РНС с типом связи многие ко многим:

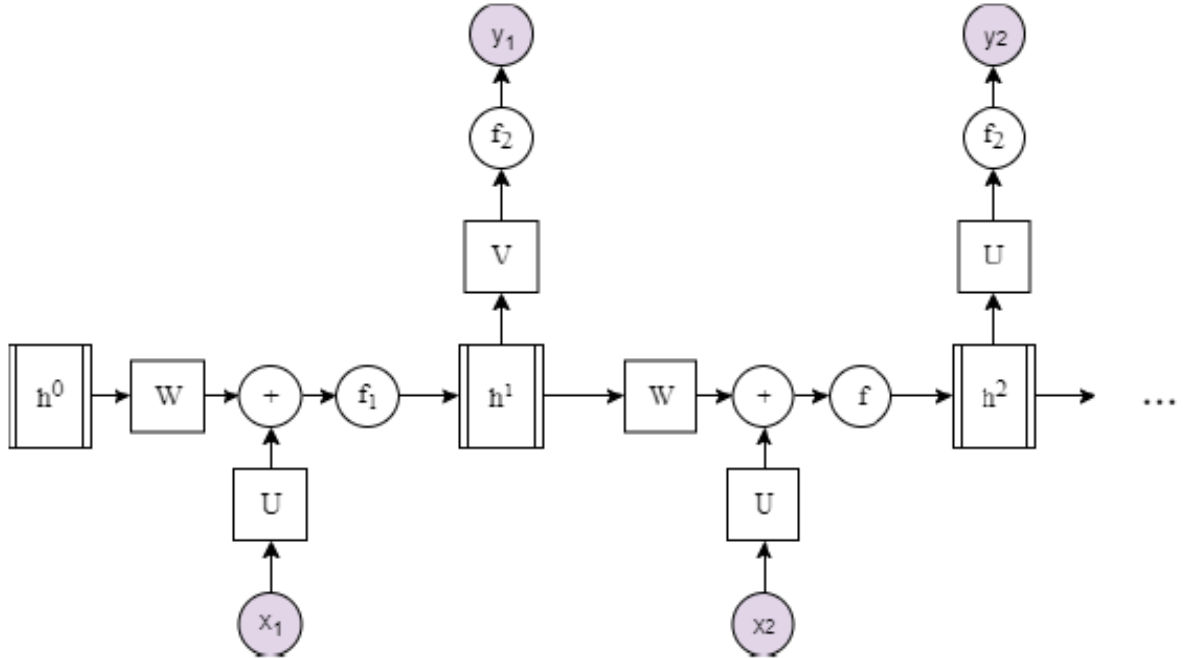


Рис. 5.3.4.1 РНС с типом связи многие ко многим, где x^i – состояние системы, x_i – входные параметры, f_1 – функция активации для скрытых блоков, f_2 – функция выхода, W , U , V – матрицы весов

Уравнения прямого распространения для РНС, изображенной на рис. 5.3.4.1:

$$a^{(t)} = Wh^{(t-1)} + Ux^{(t)} + b, \quad (5.3.4.1)$$

$$h^{(t)} = f_1(a^{(t)}), \quad (5.3.4.2)$$

$$o^{(t)} = Vh^{(t)} + c. \quad (5.3.4.3)$$

$$y^{(t)} = f_2(o^{(t)}), \quad (5.3.4.4)$$

где b и c – вектора сдвига.

Вектора сдвига необходимы, чтобы была возможность сдвигать функцию активации на какое-то пороговое значение.

Чтобы выбрать f_1 и f_2 необходимо определить класс задачи, которую необходимо решить.

Пусть заранее известны все возможные состояния динамической системы,

тогда при переходе на следующий временной шаг, необходимо выбрать одно состояние из множества состояний. Такая задача называется задачей классификации. В качестве функции активации f_1 , можно взять сигмоид (рис. 5.3.4.2) или гиперболический тангенс (рис. 5.3.4.3).

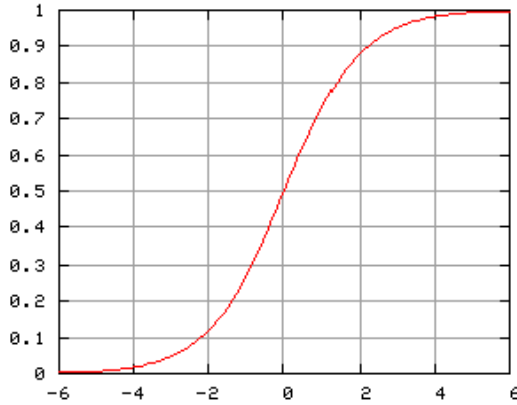


Рис. 5.3.4.2 График функции

$$\frac{1}{1+e^{-x}} - \text{сигмоид}$$

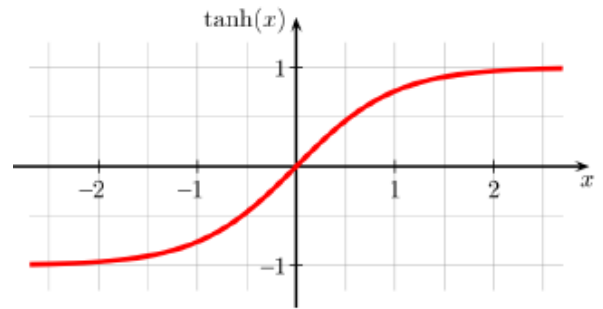


Рис. 5.3.4.3 График функции

$$\frac{2}{1+e^{-2x}} - \tanh(x)$$

Т.к. гиперболический тангенс центрирован относительно 0, то с ним легче работать: $f_1 = \tanh(a^{(t)})$.

Пусть число возможных кадров $K \geq n$, где n количество входных кадров, тогда функция f_2 должна определять вероятность принадлежности заданного x к своему классу. Такая модель является многоклассовой логистической регрессией и в качестве функции активации используется $\text{softmax}(o^{(t)}) = \frac{\exp(o_K^{(t)})}{\sum_{i=1}^K \exp(o_i^{(t)})}$.

Для выбора оптимальных параметров $\theta = [W, U, V, b, c]$ можно воспользоваться методом максимального правдоподобия.

5.3.5. МЕТОД МАКСИМИЗАЦИИ ПРАВДОПОДОБИЯ

Пусть x — некоторая случайная величина с функцией распределения $F(x, \theta)$, которая зависит от неизвестного параметра θ , а $X^n = (X_1, \dots, X_n)$ — выборка размера n , сгенерированная из распределения $F(x, \theta)$. Тогда функция правдоподобия имеет вид:

$$L(X^n, \lambda) = \prod_{i=1}^n P(X = X_i, \theta). \quad (5.3.5.1)$$

Поскольку при логарифмировании не меняются положения максимумов функции, удобно работать не с самим правдоподобием, а с логарифмом правдоподобия:

$$\ln L(X^n, \lambda) = \sum_{i=1}^n \ln P(X = X_i, \theta). \quad (5.3.5.2)$$

Оценкой максимального правдоподобия называется величина:

$$\tilde{\lambda}_{ОМП} = \arg \max_{\lambda} \ln L(X^n, \lambda). \quad (5.3.5.2)$$

5.3.6. ОПТИМИЗАЦИЯ ПАРАМЕТРОВ НЕЙРОННОЙ СЕТИ

Задача нахождения оптимальных параметров θ^* , то есть таких, при которых ошибка нейронной сети минимальна имеет вид:

$$\theta^* = \arg \min_{\theta} Q(\theta), \quad (5.3.6.1)$$

где Q – функция ошибки.

В задачах логистической регрессии в качестве функции ошибки используется кросс-энтропия, функция наибольшего правдоподобия. Пусть $\hat{y}_i = P(y = i|x)$, тогда функция кросс-энтропии выглядит так:

$$Q(\theta) = -\sum_i (\hat{y}_i \ln(y_i) + (1 - \hat{y}_i) \ln(1 - y_i)) \quad (5.3.6.2)$$

В соответствии с вышеизложенным $f_1 = \tanh(a^{(t)})$, $f_2 = \text{softmax}(o^{(t)})$, функция потерь L , необходимая для обучения – кросс-энтропия. Полученная РНС в виде графа имеет вид:

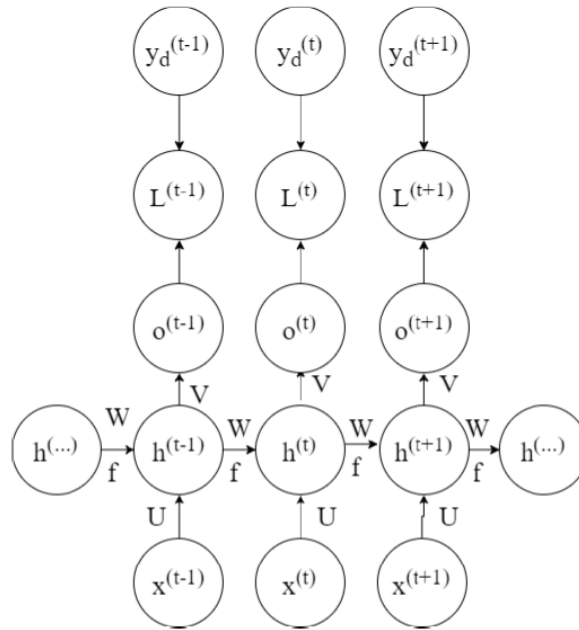


Рис. 5.3.6.1 Граф вычислений потерь при обучении рекуррентной сети. Функция потерь L измеряет насколько далеко каждый элемент отстоит от соответствующей метки y_d

Для решения задачи оптимизации (5.3.6.1) необходимо воспользоваться алгоритмом обратного распространения ошибки.

Для данной РНС алгоритм обратного распространения ошибки вычисляет:

- выражение (5.3.4.1) – прямое распространение;
- производная ошибки L по значению нейрона и его параметрам;

Другими словами: алгоритм обратного распространения заключается в вычислении произведения якобиана на градиент для каждой операции в графе:

$$\nabla_x z = \left(\frac{\partial y}{\partial x} \right)^T \nabla_y z. \quad (5.3.6.3)$$

Поскольку в РНС одни и те же параметры используются на всех временных этапах сети, градиент на каждом выходе зависит не только от расчетов текущего шага, но и от предыдущих временных шагов (рис. 5.3.6.2).

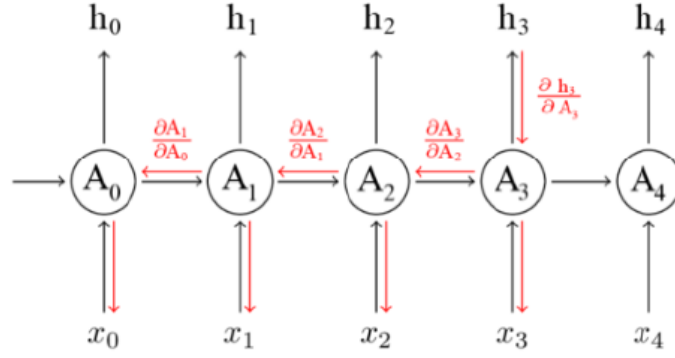


Рис. 5.3.6.2 Общий вид алгоритма обратного распространения ошибки для РНС

Исходя из рис. 5.3.6.2 в нейронной сети имеются параметры U, V, W, b, c , а также последовательность вершин, индексированных временем t : $x^{(t)}, h^{(t)}, o^{(t)}, L^{(t)}$. Для каждой вершины N необходимо рекурсивно вычислить градиент $\nabla_N L$, зная градиенты, вычисленные в вершинах, следующих за ней в графе. Рекурсия начинается с вершин, непосредственно предшествующих окончательной потере:

$$\frac{\partial L}{\partial L^{(t)}} = 1. \quad (5.3.6.4)$$

Следуя дальше по графу:

$$\nabla_{o^{(t)}} L = \frac{\partial h}{\partial h^{(t)}} \frac{\partial h^{(t)}}{\partial o^{(t)}} = \hat{y}^{(t)} - 1. \quad (5.3.6.5)$$

Двигаясь в направлении от конца последовательности к началу в последний момент времени τ у $h^{(\tau)}$ есть только один потомок $o^{(\tau)}$:

$$\nabla_{h^{(\tau)}} L = V^T \nabla_{o^{(\tau)}} L. \quad (5.3.6.6)$$

В соответствии с обратным распространением градиентов, можно совершать итерации назад во времени $t = \tau - 1$ до $t = 1$. Т.к. потомками $h^{(t)}$ для $t < \tau$ являются $o^{(t)}$ и $h^{(t+1)}$:

$$\nabla_{h^{(t)}} L = \left(\frac{\partial h^{(t+1)}}{\partial h^{(t)}} \right)^T \nabla_{h^{(t+1)}} L + \left(\frac{\partial o^{(t)}}{\partial h^{(t)}} \right)^T \nabla_{o^{(t)}} L = W^T (\nabla_{h^{(t+1)}} L) \text{diag} \left(1 - (h^{(t+1)})^2 \right) + V^T \nabla_{o^{(t)}} L, \quad (5.3.6.7)$$

где $\text{diag} \left(1 - (h^{(t+1)})^2 \right)$ – диагональная матрица с элементами $\left(1 - (h^{(t+1)})^2 \right)$.

Это якобиан функции $f_1 = \tanh$, скрытого блока i в момент времени $t + 1$.

Получив градиенты по внутренним вершинам графа вычислений, можно

получить градиенты по вершинам параметров:

$$\nabla_c L = \sum_t \left(\frac{\partial o^{(t)}}{\partial c} \right)^T \nabla_{o^{(t)}} L = \sum_t \nabla_{o^{(t)}} L, \quad (5.3.6.8)$$

$$\nabla_b L = \sum_t \left(\frac{\partial h^{(t)}}{\partial b} \right)^T \nabla_{h^{(t)}} L = \sum_t \text{diag} \left(1 - (h^{(t+1)})^2 \right) \nabla_{h^{(t)}} L, \quad (5.3.6.9)$$

$$\nabla_U L = \sum_t (\nabla_{o^{(t)}} L) h^{(t)T}, \quad (5.3.6.10)$$

$$\nabla_W L = \sum_t \text{diag} \left(1 - (h^{(t)})^2 \right) (\nabla_{h^{(t)}} L) h^{(t-1)T}, \quad (5.3.6.11)$$

$$\nabla_V L = \sum_t \text{diag} \left(1 - (h^{(t)})^2 \right) (\nabla_{h^{(t)}} L) x^{(t)T}. \quad (5.3.7.12)$$

Уравнения (5.3.6.8 – 5.3.6.12) представляют собой полную модель обучения РНС.

5.4. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ И АНАЛИЗ РЕЗУЛЬТАТОВ

В качестве входного видео создан набор данных, где змейка размером три клетки движется по полю размером 5x5. Задача нейронной сети после каждого входного изображения получить другое изображение, непротиворечащее принципам движения змейки, а также являющееся таким же изображением, как и входное на следующем временном шаге. Изображения имеют размер 25 x 25 пикселей, на вход нейронной сети изображение поступает в виде вектора [1 x 625]. Количество классов (возможных состояний змейки) $K = 94$ (рис. 5.4.1).

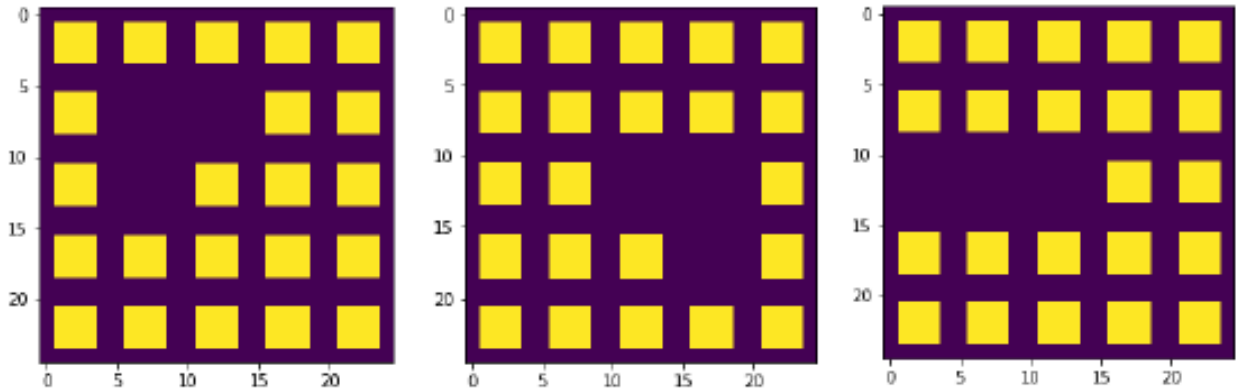


Рис. 5.4.1 Несколько из возможных состояний каскада

Нейронная сеть написана на языке программирования Python 3.7.0 с

использованием библиотеки tensorflow.

В результате нейронная сеть имеет точность $\sim 95\%$, при соотношении тестовых данных и тренировочных 1:1, результаты работы нейронной сети.

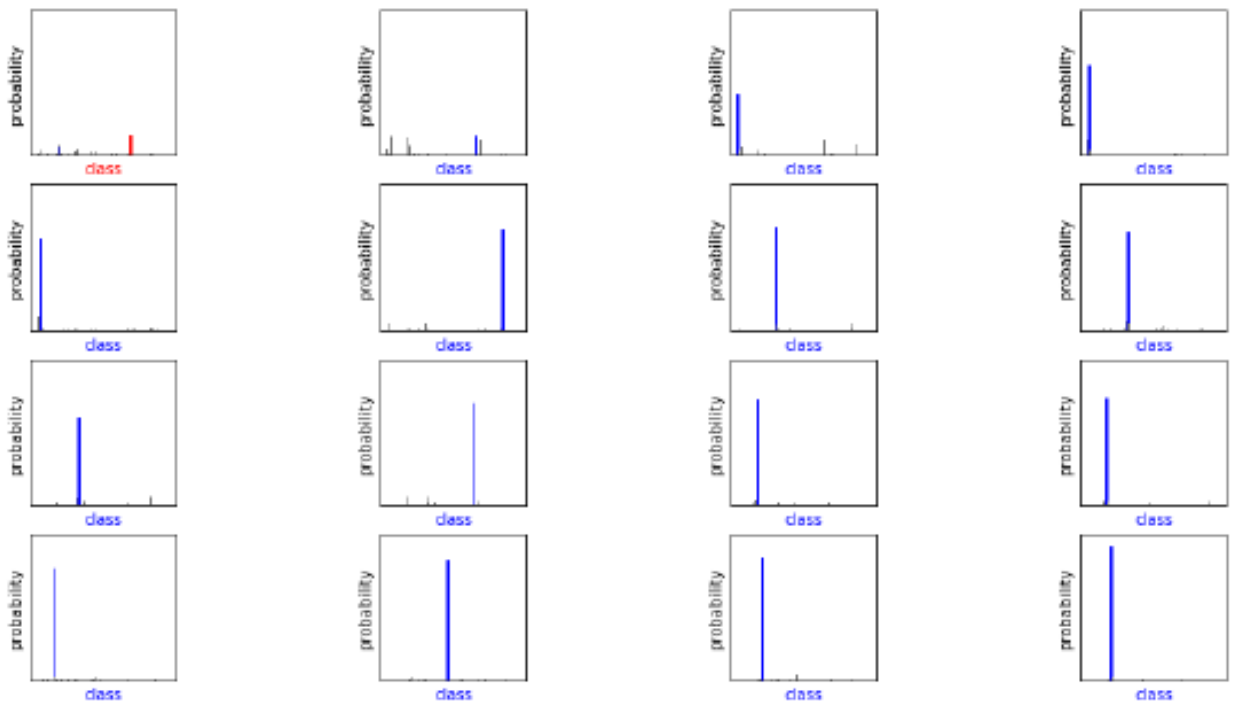


Рис. 5.4.2 Предсказания следующей итерации каскада нейронной сетью, где синим правильно предсказанное, красным – неправильное, на изображениях показано классовое распределение

В связи с отличной точностью при малом количестве обучающих данных на фоне тестовых можно сделать вывод, что рекуррентные нейронные сети могут с высокой точностью предсказывать итерации каскада.

6. НАХОЖДЕНИЕ КОГЕРЕНТНЫХ СТРУКТУР В ТУРБУЛЕНТНЫХ ПОТОКАХ

6.1.ВВЕДЕНИЕ В ТУРБУЛЕНТНЫЕ ПОТОКИ

Турбулентные течения представляют собой системы, характеризующиеся наличием хаотически распределенных и хаотически осцилирующих структур. Турбулентность – это воплощение хаоса. Долгое время развитая турбулентность считалась лишенной какого-либо порядка, однако, начиная с конца 60-х годов наметился значительный прогресс в понимании природы турбулентности, связанный с осознанием природы и структур хаоса. Стало ясно, что даже в самом

развитом турбулентном потоке существуют элементы порядка. Стоит подчеркнуть, что «истинная» турбулентность хаотична и в пространстве, и во времени.

Таким образом когерентные структуры характерны для большинства турбулентных течений и являются относительно упорядоченными структурами.

6.2.МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКОЙ СИСТЕМЫ КАК ТУРБУЛЕНТНОГО ПОТОКА

Как говорилось ранее одним из способов задания динамической системы является система дифференциальных уравнений. Для моделирования турбулентного потока можно воспользоваться уравнением Курамото-Сивашинского.

Уравнение Курамото-Сивашинского является дифференциальным уравнением в частных производных для моделирования широкого спектра нелинейных систем с внутренней неустойчивостью, таких как: распространение волн в химической реакции [27], скорость неустойчивости фронта ламинара [28], гидродинамическая турбулентность [29].

6.2.1. УРАВНЕНИЕ КУРАМОТО-СИВАШИНСКОГО

Уравнение 6.2.1.1 является представлением пространственно-временного хаоса с регулируемым параметром α .

$$\frac{\partial u}{\partial t} + \alpha u + \frac{\partial^4 u}{\partial x^4} + \frac{\partial^2 u}{\partial x^2} + u \frac{\partial u}{\partial x} = 0, \quad (6.2.1.1)$$

где $u = u(x, t)$, $x \in [0; L]$.

В соответствии с источником [30], приняв $\alpha = 0$ уравнение 6.2.1.1 будет описывать динамическую эволюцию с хаотическим решением. Для получения пространственно-временной модели турбулентности необходимо задать периодическое граничное условие и принять $L \geq 100$ [31]. В итоге уравнение Курамото-Сивашинского для моделирования турбулентного потока имеет вид:

$$\frac{\partial u}{\partial t} + \frac{\partial^4 u}{\partial x^4} + \frac{\partial^2 u}{\partial x^2} + u \frac{\partial u}{\partial x} = 0, \quad (6.2.1.2)$$

где $u = u(x, t)$, $u(x + L, t) = u(x, t)$, $0 \leq x \leq L$, $L = 100$.

Уравнение (6.2.1.2) содержит как линейную часть, так и нелинейную, что осложняет использование только явных или неявных методов. Такие задачи решаются методом IMEX (Implicit-Explicit Method). В соответствии с методом:

$$\frac{\partial u}{\partial t} = Lu + N(u), \quad (6.2.1.3)$$

где L , N – линейный и нелинейный оператор соответственно.

Пусть $Lu = -\frac{\partial^4 u}{\partial x^4} - \frac{\partial^2 u}{\partial x^2}$, а $N(u) = -u \frac{\partial u}{\partial x}$, тогда для дискретизации уравнения (6.2.1.2) по времени можно воспользоваться разностной схемой CNAB2 (CrankNicolson (Trapezoidal rule) Adams-Bashforth 2):

$$u^{n+1} = u^n + \Delta t \left[\frac{3}{2} N(u^n) - \frac{1}{2} N(u^{n-1}) \right] + \frac{\Delta t}{2} [Lu^{n+1} + Lu^n], \quad (6.2.1.4)$$

$$u^{n+1} - \frac{\Delta t}{2} Lu^{n+1} = u^n + \Delta t \left[\frac{3}{2} N(u^n) - \frac{1}{2} N(u^{n-1}) \right] + \frac{\Delta t}{2} Lu^n, \quad (6.2.1.5)$$

Для моделирования турбулентного потока необходимо задаться граничным условием вида:

$$u(x, 0) = \cos\left(\frac{x}{16}\right) \left(1 + \sin \frac{x}{16}\right), \quad (6.2.1.6)$$

Т.к. граничное условие является периодическим для дискретизации

уравнения (6.2.1.2) по пространственной переменной необходимо воспользоваться псевдоспектральным методом [32]. Суть метода заключается в использовании преобразования Фурье по пространственной переменной. Преобразование Фурье имеет вид:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(k) \cdot e^{2\pi i k x} dk. \quad (6.2.1.7)$$

Пусть $K = 2\pi k$ – волновой спектр, тогда дискретное преобразование Фурье будет иметь вид:

$$f(x) = \sum_K \hat{f}(k) \cdot e^{iKx}. \quad (6.2.1.8)$$

Применяя к (6.2.1.3) преобразование Фурье:

$$\hat{u} = \hat{L}\hat{u} + \hat{N}(\hat{u}), \quad (6.2.1.9)$$

где величины \hat{u} , \hat{L} , \hat{N} – образы в пространстве Фурье, величин u , L и N соответственно. Преобразование Фурье для линейного оператора вычисляется по формуле:

$$\hat{L}\hat{u} = K^2 - K^4. \quad (6.2.1.10)$$

Для построения преобразования Фурье нелинейного слагаемого используется следующая цепочка равенств:

$$\hat{N}(\hat{u}^2) = -\frac{1}{2} \frac{d}{dx} \hat{u}^2 = -\frac{iK}{2} \left(F \left((F^{-1}(\hat{u}))^2 \right) \right), \quad (6.2.1.11)$$

где F – дискретное преобразование Фурье.

Переписав уравнения (6.2.1.5), (6.2.1.9)-(6.2.1.11) в матричном виде, схема для решения уравнения (6.2.1.2) имеет вид:

$$\left(I - \frac{\Delta t}{2} \hat{L} \right) \hat{u}^{n+1} = \left(I + \frac{\Delta t}{2} \hat{L} \right) \hat{u}^n + \frac{3\Delta t}{2} \hat{N}^n - \frac{\Delta t}{2} \hat{N}^{n-1}, \quad (6.2.1.12)$$

где \hat{u} – вектор коэффициентов Фурье, \hat{L} – диагональная матрица. Для удобства, пусть $A = I + \frac{\Delta t}{2} \hat{L}$, $B = \left(I - \frac{\Delta t}{2} \hat{L} \right)^{-1}$, тогда уравнение (6.2.1.12) примет вид:

$$\hat{u}^{n+1} = B(A\hat{u}^n + \frac{3\Delta t}{2} \hat{N}^n - \frac{\Delta t}{2} \hat{N}^{n-1}). \quad (6.2.1.13)$$

Затем для нахождения u необходимо применить обратное преобразование Фурье

$$u = F^{-1}(\hat{u}). \quad (6.2.1.14)$$

Результат решения уравнения (6.2.1.2) представлен на рис. 6.2.1.1.

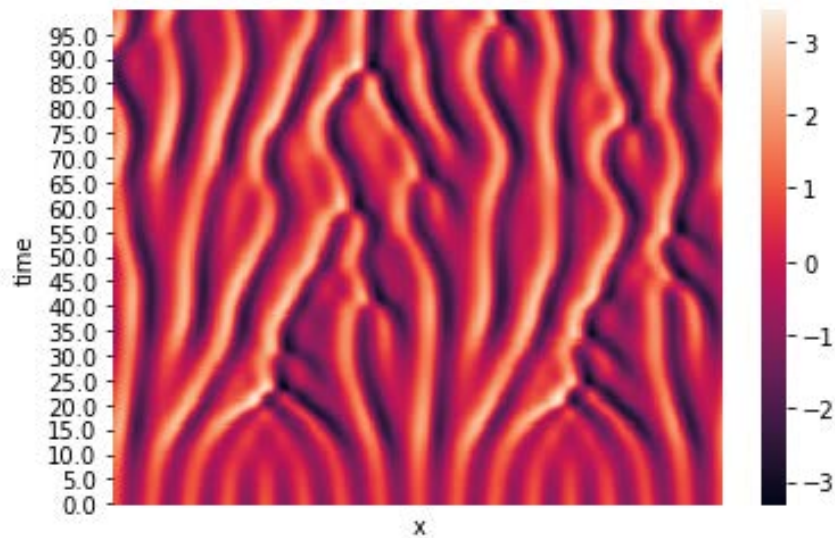


Рис. 6.2.1.1 Пространственно-временная турбулентность, смоделированная уравнением (6.2.1.2)

6.3. ИССЛЕДОВАНИЕ КОГЕРЕНТНЫХ СТРУКТУР С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ

6.3.1. ВВЕДЕНИЕ В ЗАДАЧУ

В данной работе уже рассмотрена возможность применения моделей и методов машинного обучения для исследования дискретных динамических систем. На данном этапе ставится задача демонстрации возможности применения машинного обучения для исследования непрерывных динамических систем на примере исследования когерентных структур в уравнении Курамото-Сивашинского.

Т.к. когерентные структуры в пространственно-временной турбулентности являются чем-то упорядоченным в системе хаоса и исходя из литературного обзора методов машинного обучения, можно сделать предположение, что, используя методы кластеризации, возможно выявить когерентные структуры в турбулентном потоке.

6.3.2. ВВЕДЕНИЕ В КЛАСТЕРИЗАЦИЮ

В задаче кластеризации обучающая выборка x_1, \dots, x_l состоит только из объектов, но не содержит ответы на них, а также одновременно является и тестовой выборкой. Требуется расставить метки y_1, \dots, y_l таким образом, чтобы похожие друг на друга объекты имели одинаковую метку, то есть разбить все объекты на некоторое количество групп.

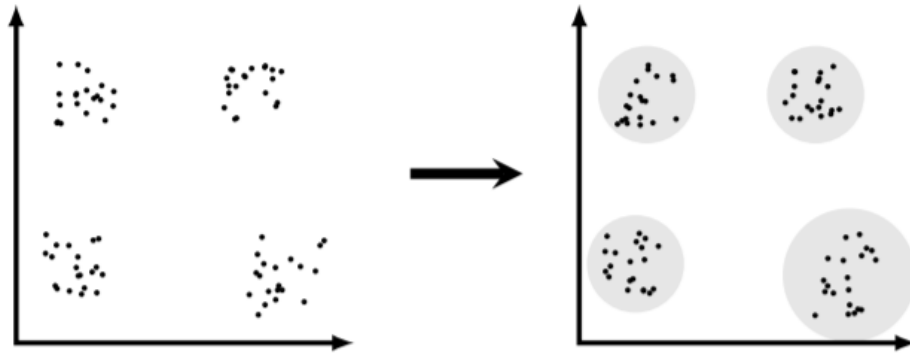


Рис. 6.3.2.1 Пример задачи кластеризации

Задачу кластеризации также можно представить как восстановление отображения $x \rightarrow y$.

В зависимости от специфики задачи и особенностей используемых в задаче данных задачи кластеризации могут сильно отличаться. Может, например, отличаться форма, размер и иерархия кластеров, а также тип задачи (основная или побочная) и тип классификации (жесткая или мягкая).

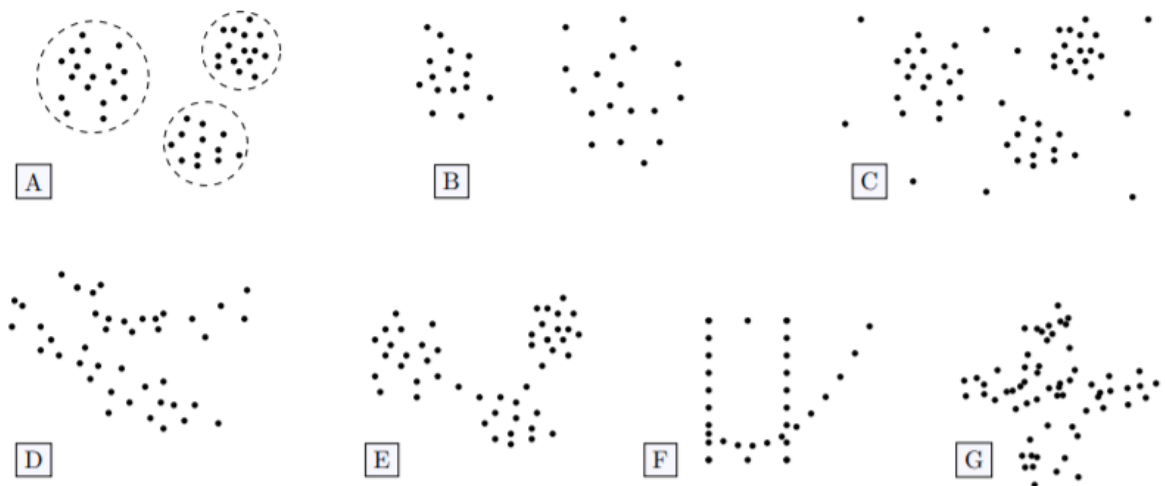


Рис. 6.3.2.2 Пример различных форм кластеров

Кластеры могут иметь совершенно различную форму.

А. В простейшем случае кластеры представляют собой «сгустки точек» и могут быть легко выделены окружностью.

- В. Иногда кластеры принимают более сложную форму, но все также могут быть легко выделены.
- С. Возможен случай, когда между кластерами есть такие точки, которые сложно отнести к определенному кластеру.
- Д. Кластеры могут представлять собой вытянутые ленты. В этом случае можно найти пару точек из разных кластеров, которые находятся ближе некоторой пары из одного кластера. Такие кластеры можно выделить, добавляя к строящемуся кластеру ближайшую точку.
- Е. Кластеры могут плавно перетекать друг в друга. В этом случае описанная для предыдущего случая стратегия уже не работает.
- Ф. Кластеры могут быть образованы по некоторому закону, который все же не известен.
- Г. Кластеры могут пересекаться. В этом случае достаточно сложно определить, к какому кластеру относятся некоторые объекты.

Поскольку кластеры могут иметь различный вид в зависимости от специфики задачи, универсального алгоритма кластеризации не существует. При этом использование различных алгоритмов кластеризации в одной и той же задаче может давать совершенно разный ответ. Поэтому необходимо рассмотреть несколько основных алгоритмов кластеризации.

6.3.3. МЕТОД К-СРЕДНИХ

Простейшим методом кластеризации является метод k -средних, где k — число кластеров, которые требуется выделить. Предполагается, что число k известно.

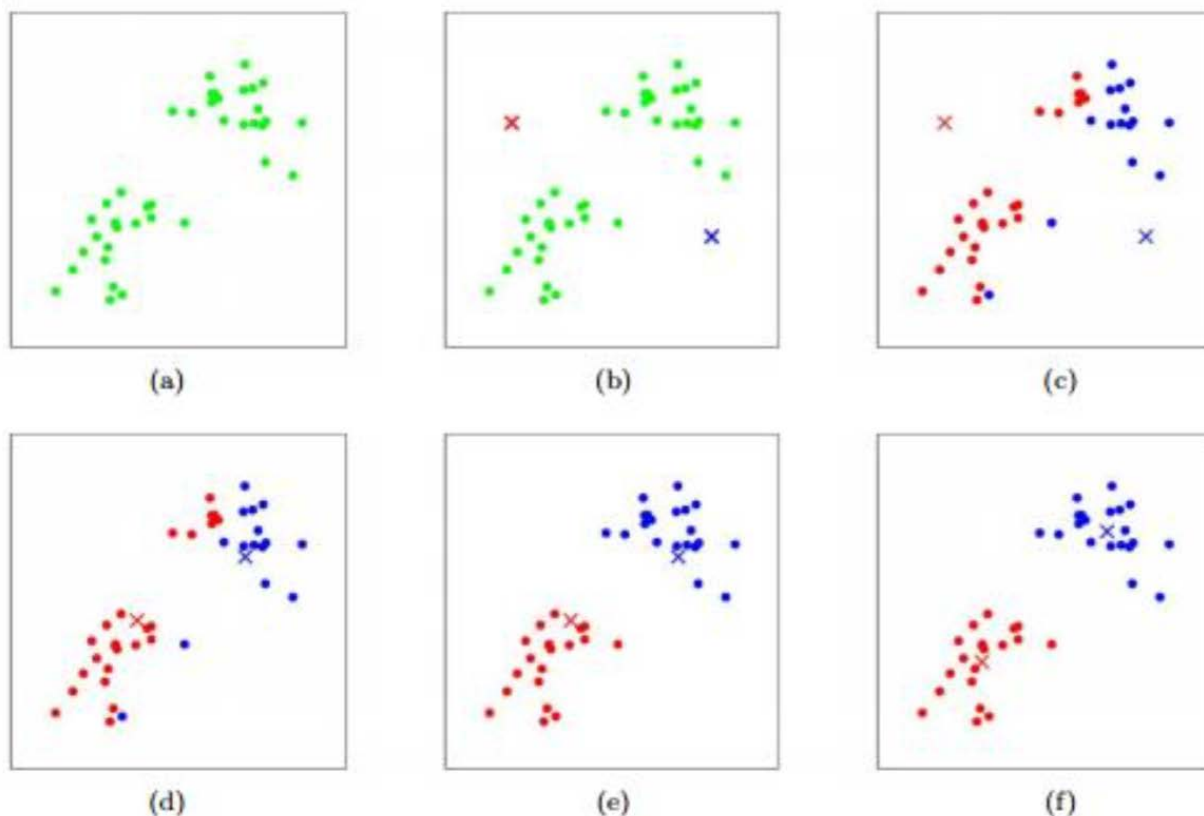


Рис. 6.3.3.1 Работа алгоритма k-средних

Алгоритм k-средних представляет собой итеративный процесс. Пусть в начале произвольным образом выбраны центры классов. Объект относится к тому кластеру, расстояние до центра которого меньше. На каждой итерации сначала пересчитываются положения центра каждого класса как среднее арифметическое отнесенных к нему точек, а после этого объекты перераспределяются на основании новых положений центров.

Среднее внутрикластерное расстояние имеет следующий вид:

$$F_0 = \frac{\sum_{i < j} [y_i = y_j] p(k_i, x_j)}{\sum_{i < j} [y_i = y_j]} \rightarrow \min. \quad (6.3.3.1)$$

6.3.4. EXPECTATION MAXIMIZATION (ЕМ-АЛГОРИТМ)

ЕМ-алгоритм, как и k-средних, является итеративным методом: сначала задается начальное приближение, а затем на каждом шаге форма кластеров уточняется.

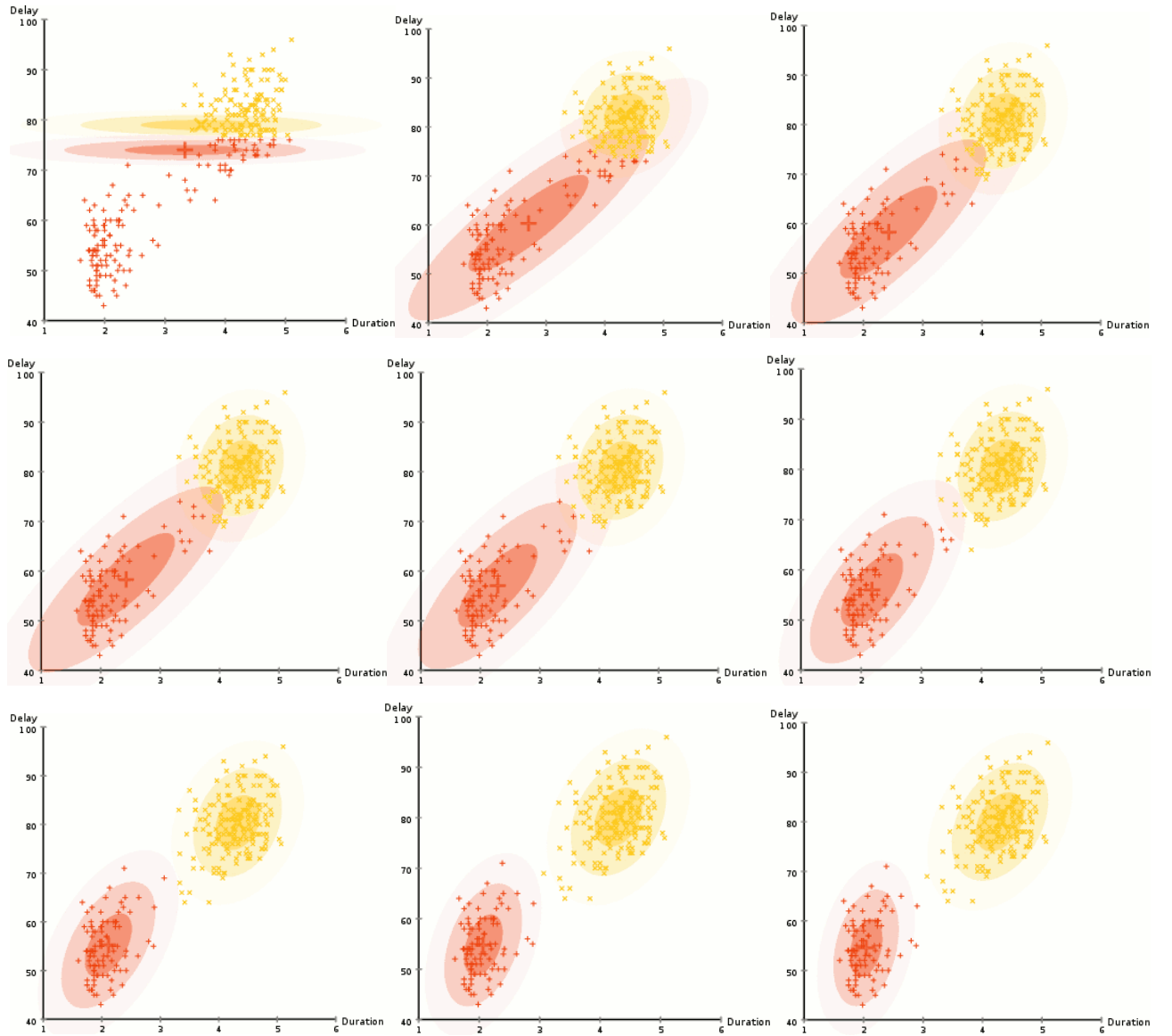


Рис. 6.3.4.1 Пример работы ЕМ-алгоритма

Пусть $L(\theta; X, Z) = p(X, Z|\theta)$ — функция правдоподобия. Тогда в ЕМ-алгоритме на Е-шаге происходит построение функции ожидаемого значения логарифма правдоподобия, зависящей от параметров θ :

$$Q(\theta|\theta^{(t)}) = E_{Z|X, \theta^{(t)}} \log L(\theta; X, Z), \quad (6.3.4.1)$$

а на М-шаге вычисляются значения параметров $\theta^{(t+1)}$, которые максимизируют $Q(\theta|\theta^{(t)})$:

$$\theta^{(t+1)} = \arg \max_{\theta} Q(\theta|\theta^{(t)}). \quad (6.3.4.2)$$

Другими словами, суть ЕМ-алгоритма заключается в том, что для построения итерационного процесса нужно ввести скрытые переменные таким образом, чтобы вычисления были как можно более простые. Оказывается, что в

качестве таких скрытых переменных удобно рассмотреть:

$$g_{ji} = P(j|x_i) = \frac{\omega_j p_j(x_i)}{\sum_{k=1}^K \omega_k p_k(x_i)}, \quad (6.3.4.3)$$

где $\omega_1, \dots, \omega_K$ — априорные вероятности кластеров, $p_1(x), \dots, p_K(x)$ — плотности распределения кластеров.

На втором шаге (М-шаге) решается задача максимизации правдоподобия с зафиксированными $P(j|x_i)$. Если приравнять производные по параметрам к нулю, получаются выражения:

$$\omega_j = \frac{1}{N} \sum_{i=1}^N g_{ji}, \quad \theta_j = \arg \max_{\theta} \sum_{i=1}^N g_{ji} \ln \varphi(\theta; x), \quad (6.3.4.4)$$

где $p_j(x) = \varphi(\theta_j; x)$.

6.3.5. АГЛОМЕРАТИВНАЯ ИЕРАРХИЧЕСКАЯ КЛАСТЕРИЗАЦИЯ

Иерархическая кластеризация — кластеризация, в которой кластеры получаются вложенными друг в друга. Выделяют два способа это сделать:

- агломеративный подход: каждый объект помещается в свой собственный кластер, которые постепенно объединяются;
- дивизивный подход: сначала все объекты помещаются в один кластер, который затем разбивается на более мелкие кластеры;

На данном примере показан ход выполнения агломеративной иерархической кластеризации. Считается, что выбран некоторый способ вычисления расстояния между кластерами.

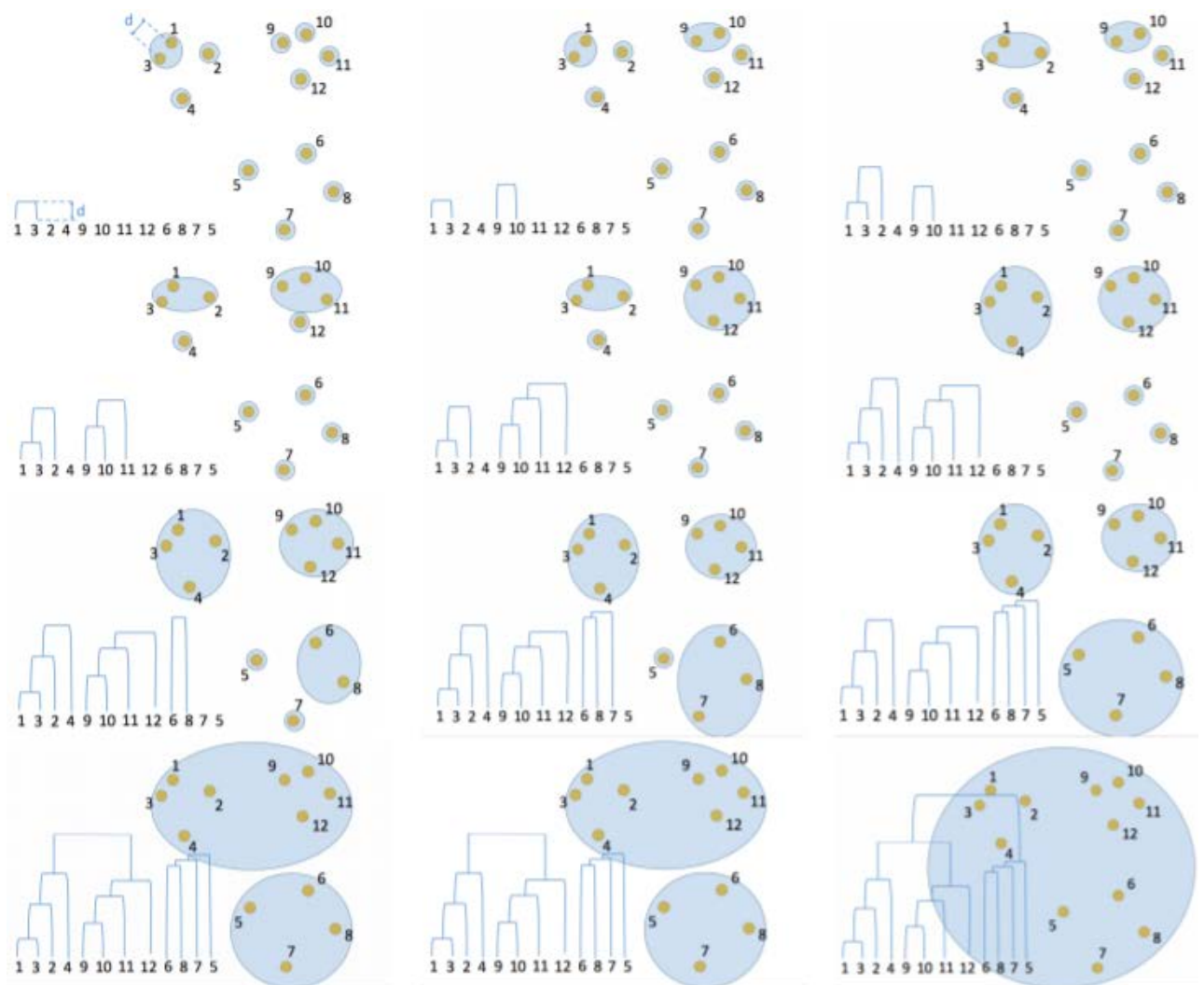


Рис. 6.3.5.1 Процесс агломеративной иерархической кластеризации вместе с дендограммой

Расстояние между кластерами можно ввести несколькими разными способами.

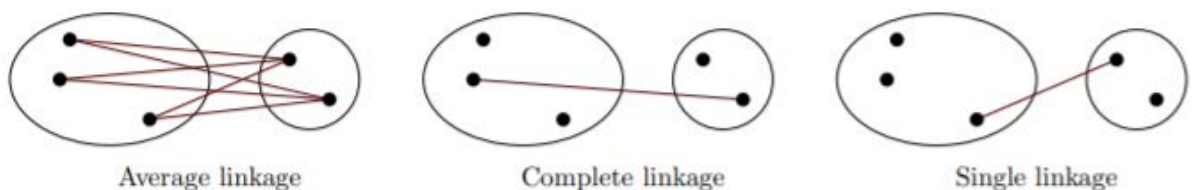


Рис. 6.3.5.2 Различные способы ввести расстояние между кластерами

С помощью формулы Ланса-Уильямса можно обобщить множество разных способов ввести расстояния между кластерами:

$$R(U \cup V, S) = \alpha_U R(U, S) + \alpha_V R(V, S) + \beta R(U, V) + \gamma |R(U, S) - R(V, S)| \quad (6.3.5.1)$$

Эта формула выражает расстояние между кластером, которым получается

в результате слияния двух кластеров U и V и каким-то третьим кластером S . Формула позволяет рекурсивно получить расстояние между двумя сложными кластерами, если известно расстояние между более простыми. Разные коэффициенты в этой формуле приводят к разным способам вычислять расстояние между кластерами, в том числе:

- расстояние ближайшего соседа (при $\alpha_U = \alpha_V = \frac{1}{2}$, $\beta = 0$, $\gamma = -\frac{1}{2}$):

$$R^B(W, S) = \min_{\omega \in W, s \in S} \rho(\omega, s); \quad (6.3.5.2)$$

- расстояние дальнего соседа (при $\alpha_U = \alpha_V = \frac{1}{2}$, $\beta = 0$, $\gamma = \frac{1}{2}$):

$$R^D(W, S) = \max_{\omega \in W, s \in S} \rho(\omega, s); \quad (6.3.5.3)$$

- среднее расстояние (при $\alpha_U = \frac{|U|}{|W|}$, $\alpha_V = \frac{|V|}{|W|}$, $\beta = \gamma = 0$):

$$R^C(W, S) = \frac{1}{|W||S|} \sum_{\omega \in W} \sum_{s \in S} \rho(\omega, s); \quad (6.3.5.4)$$

- расстояние между центрами кластеров (при $\alpha_U = \frac{|U|}{|W|}$, $\alpha_V = \frac{|V|}{|W|}$, $\beta = -\alpha_U \alpha_V$, $\gamma = 0$):

$$R^H(W, S) = \rho^2 \left(\sum_{\omega \in W} \frac{\omega}{|W|}, \sum_{s \in S} \frac{s}{|S|} \right); \quad (6.3.5.5)$$

6.3.6. МЕТОД ОСНОВАННЫЙ НА ПЛОТНОСТИ

Идея методов, основанных на плотности, заключается в том, чтобы рассматривать плотность точек в окрестности каждого объекта выборки. Если в окрестности радиуса R с центром в некоторой точке выборки находится N или более других точек выборки, то такая точка считается основной. Здесь R и N — параметры алгоритма. Если точек меньше, чем N , но в окрестности рассматриваемой точки содержится основная точка, то такая точка называется граничной. В ином случае точка считается шумовой.

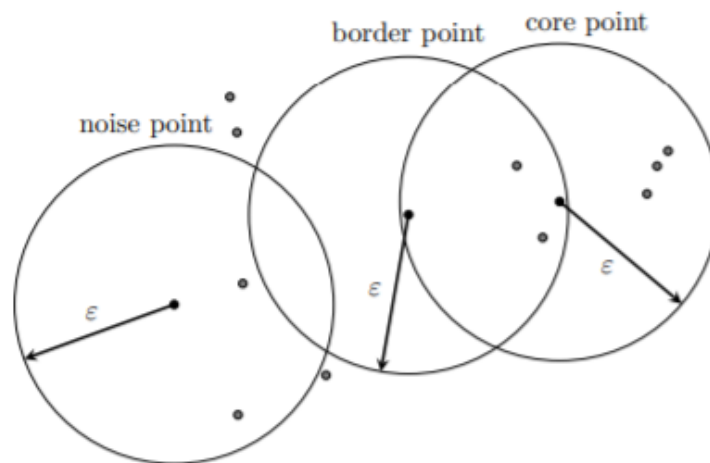


Рис. 6.3.6.1 Принцип работы алгоритма DBSCAN: изображены основная, пограничная и шумовая точки

DBSCAN — это один из таких методов, который состоит из следующих шагов:

1. разделить точки на основные, пограничные и шумовые;
2. отбросить шумовые точки;
3. соединить основные точки, которые находятся на расстоянии ϵ друг от друга;
4. каждую группу соединенных основных точек объединить в свой кластер (то есть выделить связные компоненты в получившемся графе);
5. отнести пограничные точки к соответствующим им кластерам;



Рис. 6.3.6.2 Принцип работы алгоритма DBSCAN

Как видно на последнем примере, DBSCAN хорошо справляется с нетривиальными формами кластеров и успешно отделяет шумовые точки, которые могли бы сильно испортить работу других алгоритмов кластеризации.

6.3.7. ИСПОЛЬЗОВАНИЕ МЕТОДОВ КЛАСТЕРИЗАЦИИ ДЛЯ ИССЛЕДОВАНИЯ КОГЕРЕНТНЫХ СТРУКТУР

Прежде чем применять методы кластеризации нужно проверить наличие кластерной структуры. Для этого можно сгенерировать p случайных точек из равномерного распределения и p случайных точек из обучающей выборки, в данном случае из набора точек, полученных в результате решения уравнения (6.2.1.2). После этого необходимо вычислить статистику Хопкинса:

$$H = \frac{\sum_{i=1}^P \omega_i}{\sum_{i=1}^P u_i + \sum_{i=1}^P \omega_i}, \quad (6.3.5.5)$$

где ω_i – расстояние от i -ой случайной точки до ближайшей случайной, u_i – расстояние от i -ой точки из выборки до другой ближайшей точки из выборки. Если статистика получается близкой к $\frac{1}{2}$, это значит, что выборка более-менее равномерно заполняет пространство признаков. Если же статистика получается вблизи нуля, это означает, что точки как-то группируются.

Статистика Хопкинса для полученных данных из уравнения (6.2.1.2) составляет $\sim 0,159$, что говорит о наличии кластерной структуры.

Как говорилось ранее, нет универсального метода кластеризации, следовательно, необходимо применить различные методы для выявления когерентных структур.

Для применения метода k -средних необходимо задаться количеством классов. Чтобы выбрать оптимальный параметр k можно воспользоваться коэффициентом силуэта.

Коэффициент силуэта — метрика качества, которая позволяет выбрать количество кластеров. Коэффициент силуэта для некоторого фиксированного объекта определяется следующим образом:

$$s = \frac{b-a}{\max(a,b)}, \quad (6.3.5.5)$$

где a - среднее расстояние от данного объекта до других объектов из того же кластера, b – среднее расстояние от данного объекта до объектов из ближайшего другого кластера.

Обычно коэффициент силуэта положителен, но, вообще говоря, может меняться в пределах от -1 до 1 . Если объект находится вблизи границы кластера, и близко к нему расположен другой небольшой кластер, коэффициент силуэта для этого объекта получится очень маленьким. Обычно вычисляют среднее значение коэффициента силуэта, а также для каждого кластера строят график, показывающий количество точек с различными значениями коэффициента силуэта. Среднее значение на таких графиках отмечается пунктирной линией.

Продемонстрировать данный метод можно на сгенерированной выборке с заранее известным число кластеров равным четырем с использованием метода k -средних (рис.6.3.7.1).

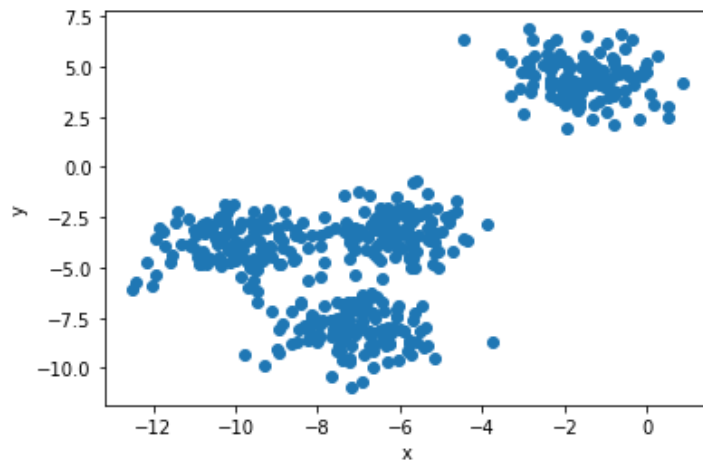


Рис. 6.3.7.1 Сгенерированные данные из четырех распределений.

В случае двух кластеров в обоих из них есть точки, коэффициент силуэта для которых больше его среднего значения по всей выборке. Также разброс значений по кластерам не очень большой.

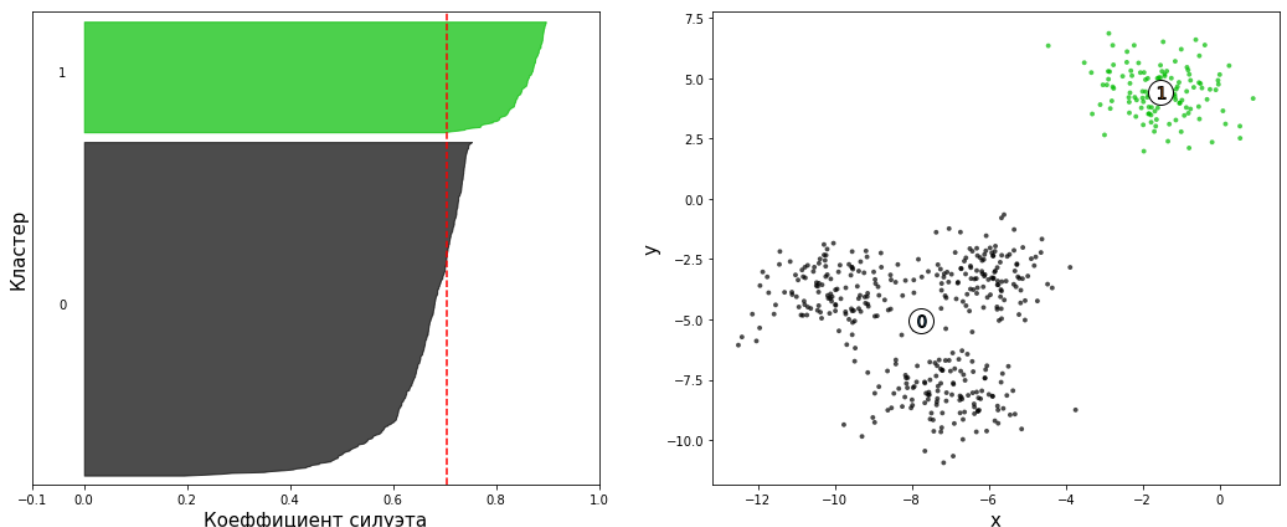


Рис. 6.3.7.2 График для анализа коэффициента силуэта(слева), результат кластеризации методом k -средних на два кластера(справа)

В случае трех кластеров разброс значений коэффициента силуэта в различных кластерах уже большой. При этом в третьем кластере все значения оказались меньше среднего значения по всей выборке. Такая кластеризация выглядит не очень убедительно и такое число кластеров лучше не выбирать.

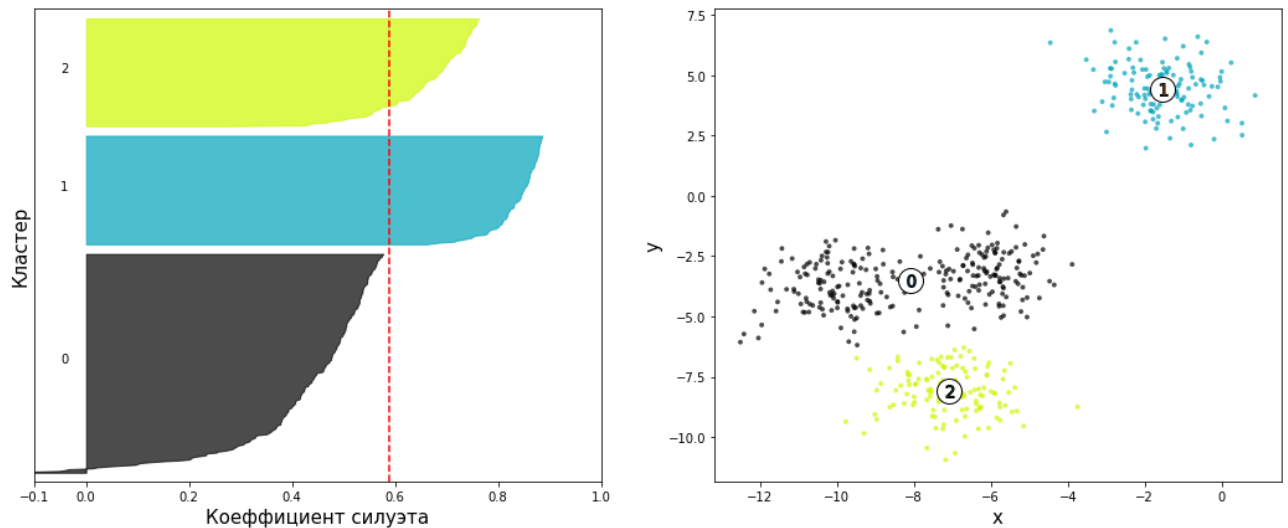


Рис. 6.3.7.3 График для анализа коэффициента силуэта(слева), результат кластеризации методом k -средних на три кластера(справа)

В случае четырех кластеров разброс значений коэффициента силуэта в различных кластерах небольшой, а также во всех кластерах есть точки, значения коэффициентов силуэта в которых больше среднего значения по выборке. Кластеризация с таким числом кластеров выглядит очень хорошо, в чем можно убедиться по визуализации на выборке.

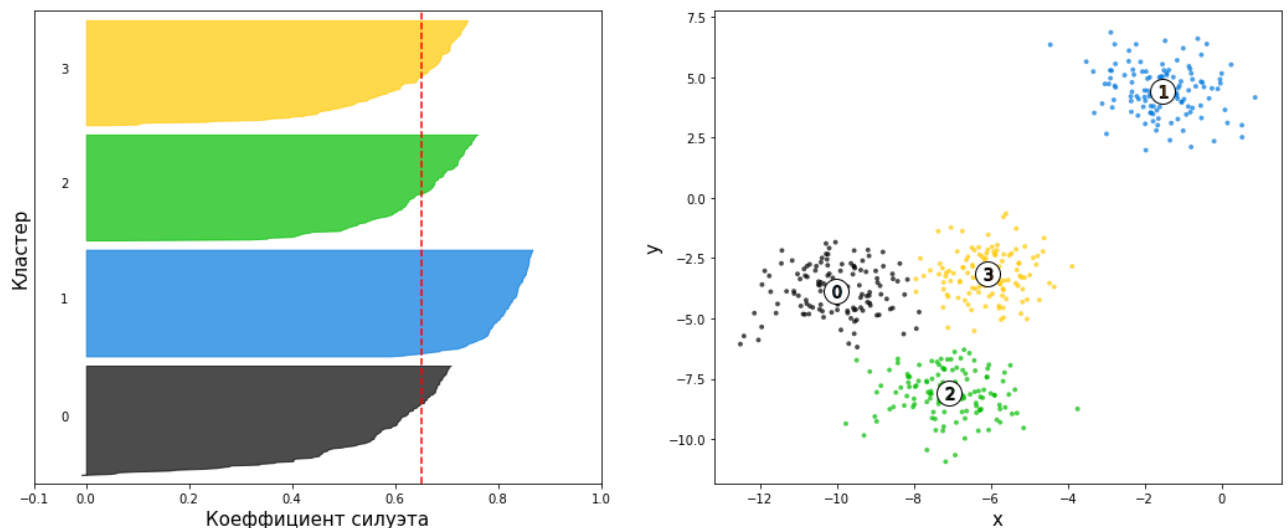


Рис. 6.3.7.4 График для анализа коэффициента силуэта(слева), результат кластеризации методом k -средних на четыре кластера(справа)

Результат анализа данных, полученных из уравнения (6.2.1.2), с помощью метода k -средних, представлен на рис.6.3.7.5. Оптимальное количество кластеров, найденное с помощью коэффициента силуэта равно двум.

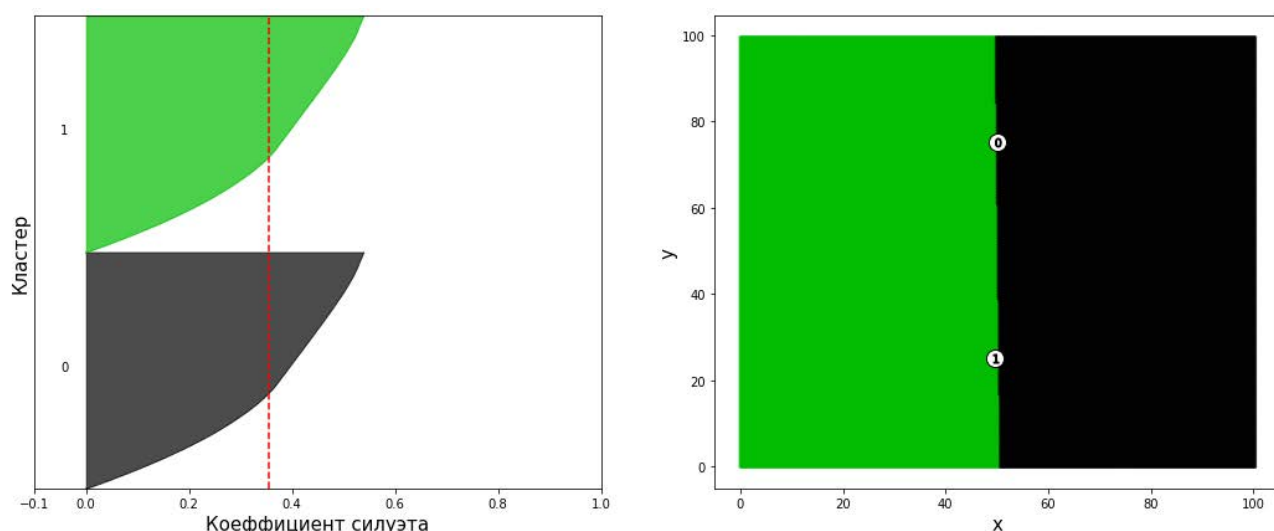


Рис. 6.3.7.5 Результат выделения когерентных структур представлен справа, в качества метода кластеризации используется алгоритм k -средних, с количеством кластеров равным двум

Из рис. 6.3.7.5 можно сделать вывод, что метод k -средних не подходит для данной задачи, т.к. структура данных очень сложная. Исходя из пункта 6.3.6 метод DBSCAN хорошо справляется с кластерами сложной формы.

Чтобы определить оптимальный параметр ε метода DBSCAN, имеет смысл построить график, по оси y у которого отложено расстояние до k -го соседа, а по оси x — количество точек, расстояние до k -го соседа соседа у которых меньше.

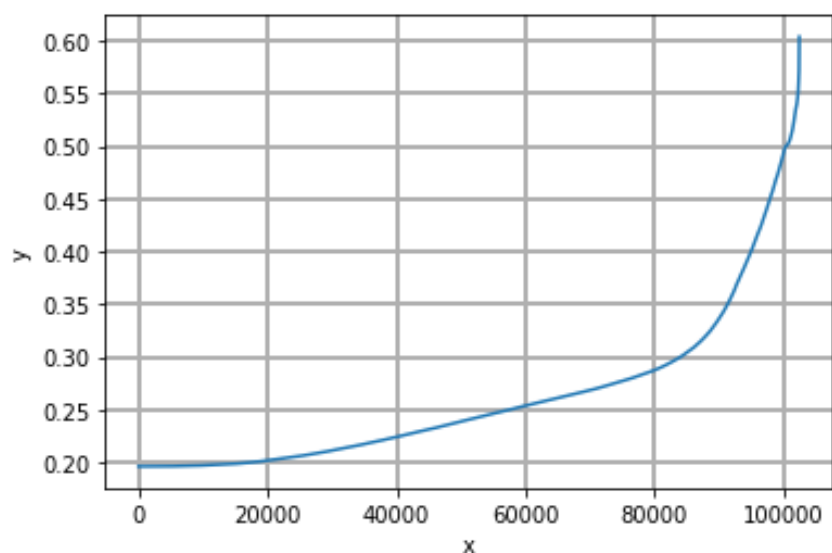


Рис. 6.3.7.6 График для определения оптимального ε для данных, полученных из уравнения (6.2.1.2)

Начиная с некоторого номера на этом графике происходит резкий рост расстояния, а значит можно естественным способом выбрать расстояние, выбираемое в качестве радиуса ε окрестности. В соответствии с графиком 6.3.7.6 оптимальное значение $\varepsilon = 0.3$. Чтобы показать значимость выбора оптимального значения параметра ε проведена кластеризация методом DBSCAN с $\varepsilon = 0.3$ и $\varepsilon = 0.5$.

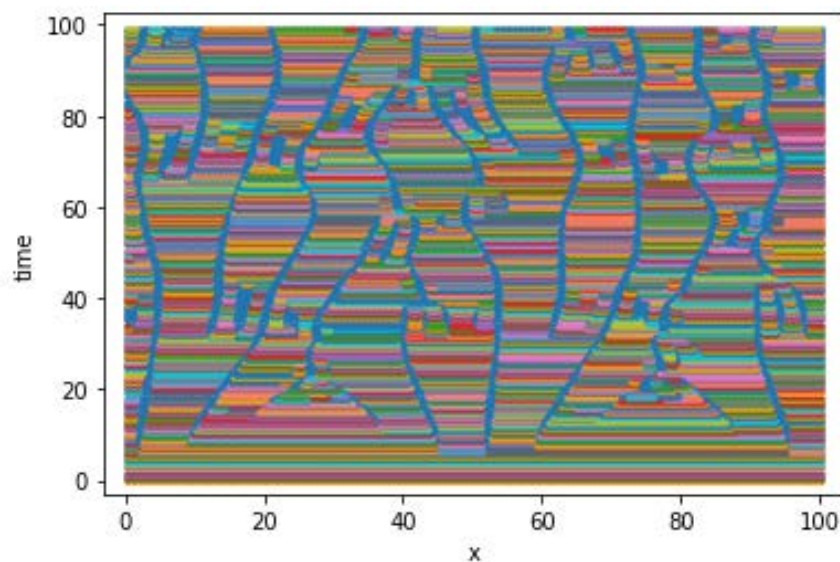


Рис. 6.3.7.7 Результат кластеризации методом DBSCAN при $\varepsilon = 0.3$

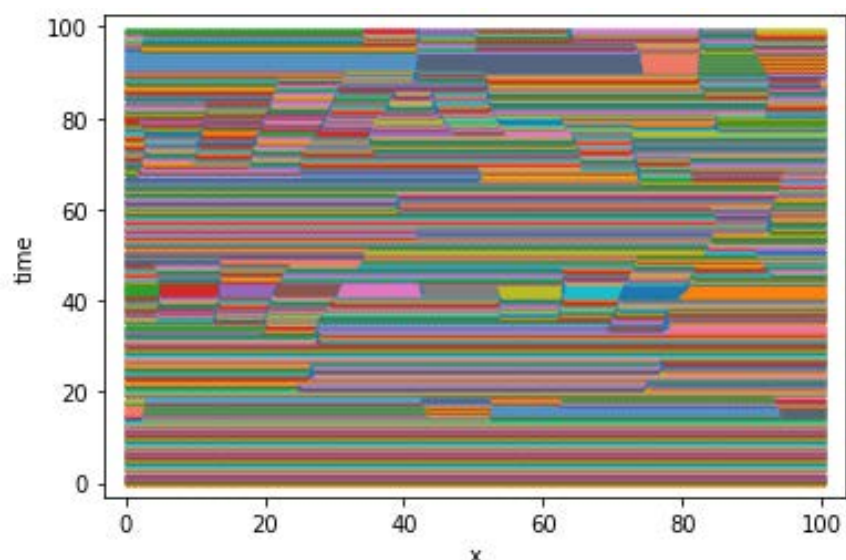


Рис. 6.3.7.8 Результат кластеризации методом DBSCAN при $\varepsilon = 0.5$

Проанализировав рис.6.3.7.7 и рис.6.3.7.8, можно сделать вывод, о том что при использовании метода DBSCAN, выбрав оптимальный параметр ε , выделены кластеры характеризующие пространственный хаос, а также отдельный кластер(синий) с наличием когерентных структур.

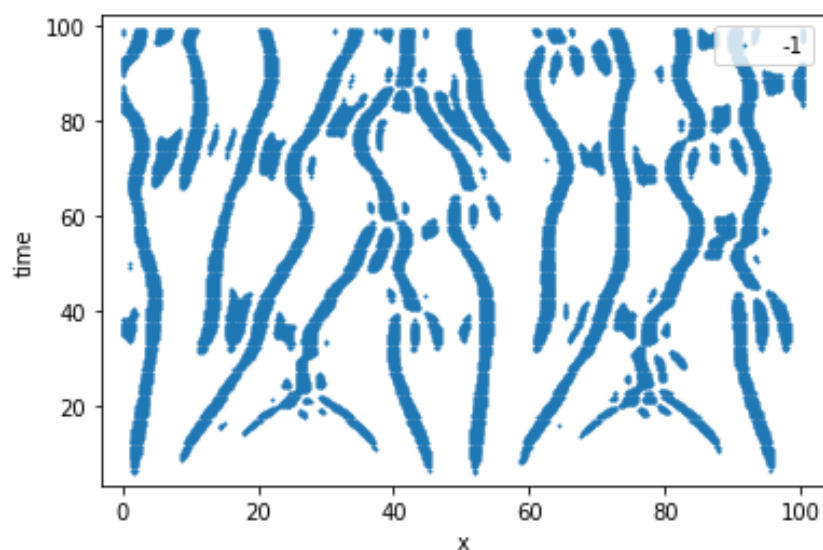


Рис. 6.3.7.9 Кластер №-1, выделенный методом DBSCAN

В результате первичной кластеризации удалось убрать большое количество данных, не соответствующих когерентной структуре.

Т.к. метод DBSCAN хорошо работает с такими данными, предлагается провести еще одну кластеризацию этим методом. Исходя из рис.6.3.7.10 оптимальный параметр $\varepsilon = 0.55$, в результате получилась следующая кластеризация, изображенная на рис.6.3.7.11.

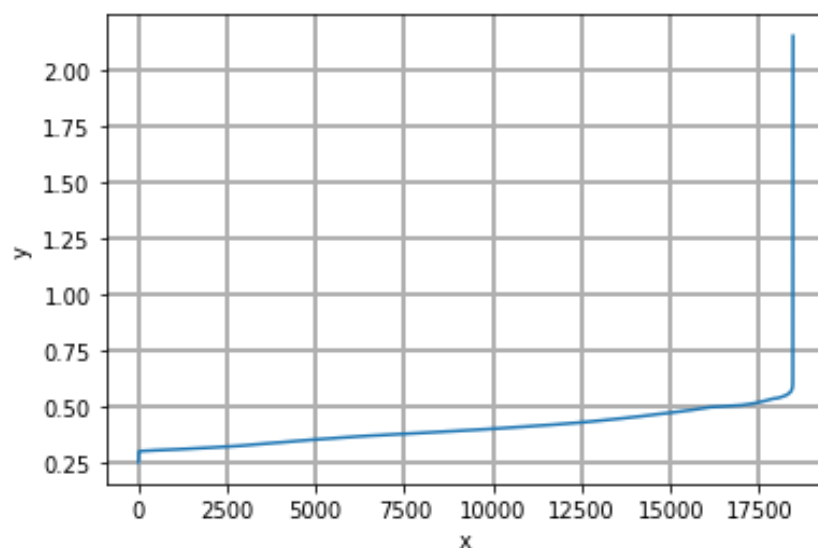


Рис. 6.3.7.10 График для определения оптимального ε для данных, полученных в результате первичной кластеризации

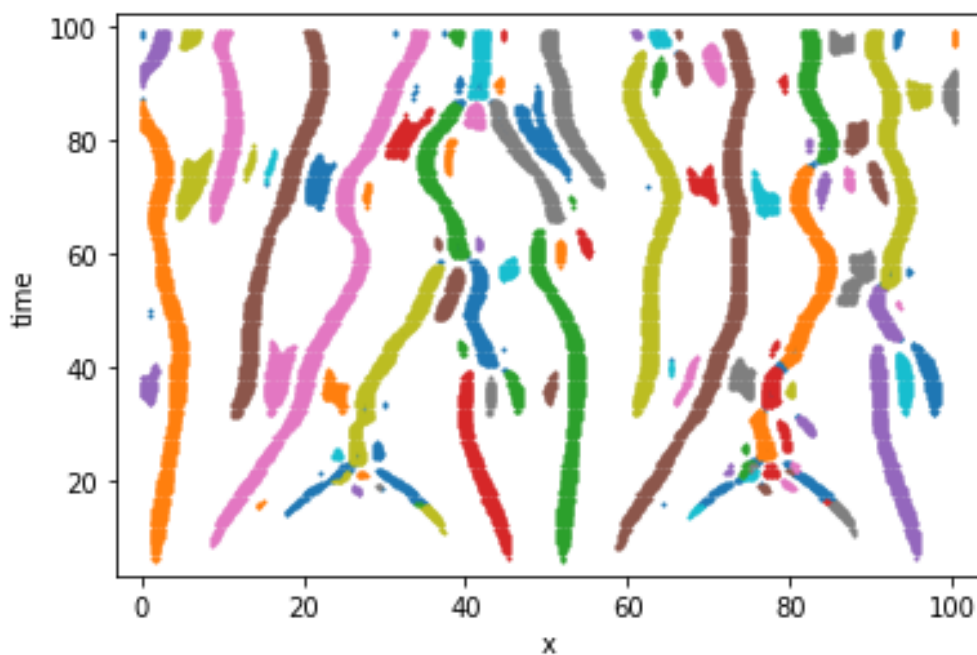


Рис. 6.3.7.11 Результат повторной кластеризации методом DBSCAN при $\varepsilon = 0.55$

В результате повторной кластеризации методом DBSCAN не удалось выделить когерентные структуры. Для повторной кластеризации предлагается использовать ЕМ-алгоритм.

В качестве параметров ЕМ-алгоритма используется тип ковариации и количество кластеров. Матрица ковариаций определяет направления и длины осей контуров плотности распределения. Типы ковариации бывают:

- полная, означает, что компоненты могут независимо принимать любую

позицию и форму;

- связанная, означает, что они имеют одинаковую форму, но форма может быть любой;
- диагональная, означает, что оси контура ориентированы вдоль координат;
- сферическая это диагональная ситуация с круговыми контурами

Для выбора оптимальных параметров можно воспользоваться Байесовским информационным критерием:

$$BIC = k \ln n - 2 \ln L, \quad (6.3.5.5)$$

где L — максимальное значение функции правдоподобия построенной модели, k — количество параметров в модели (оцененных по методу максимального правдоподобия).

Суть данного критерия в том, что в нем выбирается модель с большей апостериорной вероятностью, далее используется разложение в значении второго порядка через информационную матрицу Фишера, и потом оценка интеграла от экспоненты методом Лапласа. Соответственно, лучшая модель имеет минимальное значение Байесовского критерия информативности.

На рис.6.3.7.12 продемонстрирован пример выбора модели ЕМ-алгоритма на основе Байесовского критерия информативности для кластеризации данных на рис.6.3.7.1.

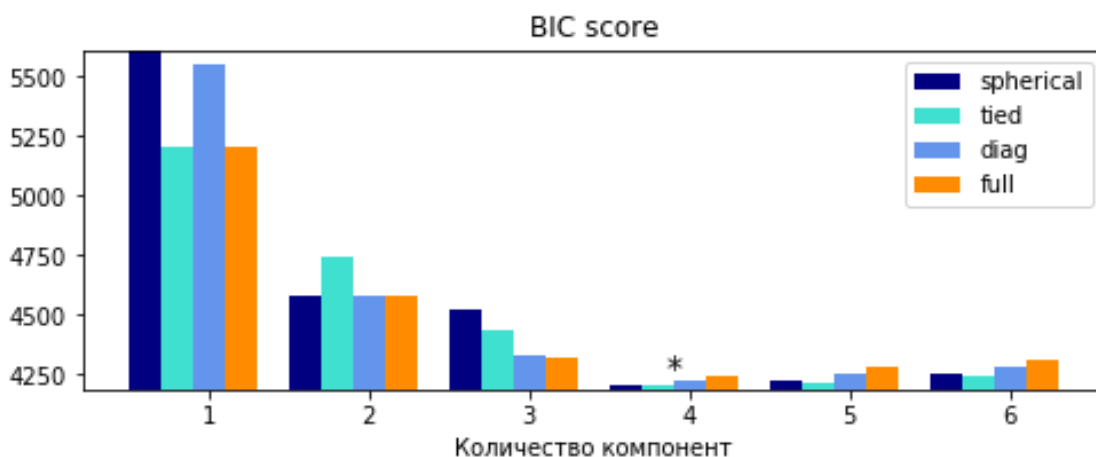


Рис. 6.3.7.12 Выбор модели на основе Байесовского критерия информативности

В соответствии с Байесовским критерием информативности оптимальная

модель EM-алгоритма для данных из рис.6.3.7.1 имеет связанный тип ковариации и количество кластеров равным четырем. На рис.6.3.7.13 изображен результат кластеризации с помощью данной модели.

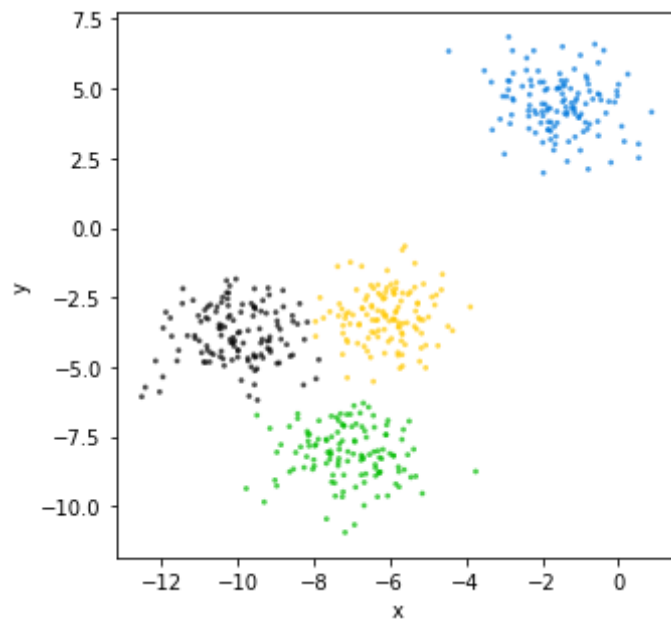


Рис. 6.3.7.13 Результат работы EM-алгоритма

Используя Байесовский критерий информативности для получения оптимальных параметров модели EM-алгоритма для кластеризации данных, полученных после первичной кластеризации, сделан вывод, что тип ковариации – сферический, число кластеров – 30. В результате получается следующая кластеризация:

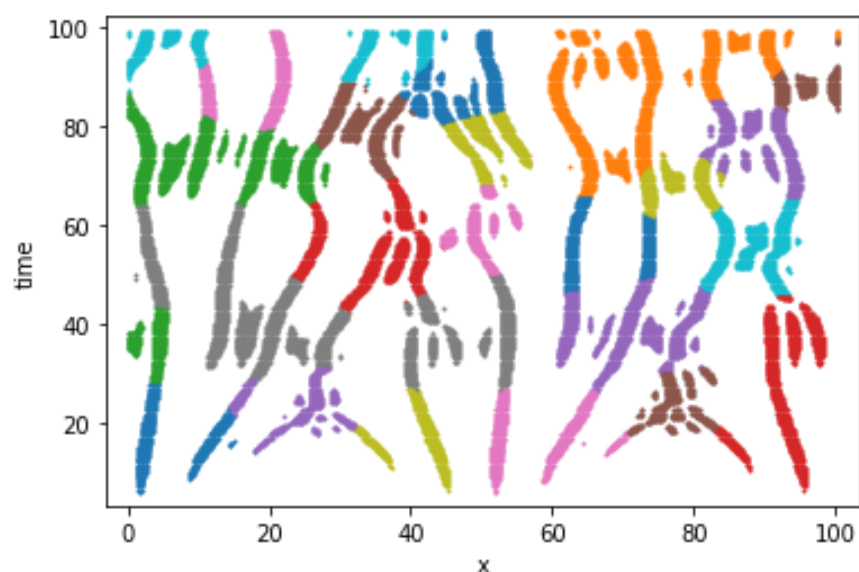


Рис. 6.3.7.13 Результат выделения когерентных структур для данных, полученных решением уравнения (6.2.1.2)

6.4. АНАЛИЗ РЕЗУЛЬТАТОВ

В результате применения методов кластеризации с выбором оптимальных параметров с целью выделения когерентных структур в пространственно-временной турбулентности, смоделированной уравнением Курамото-Сивашинского, получен результат представленный на рис.6.3.7.13.

На рис.6.4.1 изображены истинные когерентные структуры в турбулентном потоке.

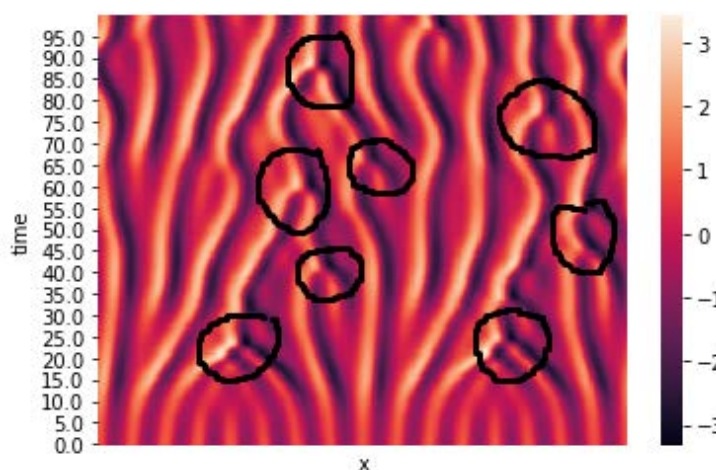


Рис. 6.4.2 Пространственно-временная турбулентность, смоделированная уравнением (6.2.1.2), где черными кругами выделены когерентные структуры

Выделив те же зоны на рис.6.3.7.13, можно сделать вывод о том, что с помощью методов кластеризации удалось выделить структуры схожие с когерентными, т.к. некоторые кластеры практически точно соответствуют когерентным структурам.

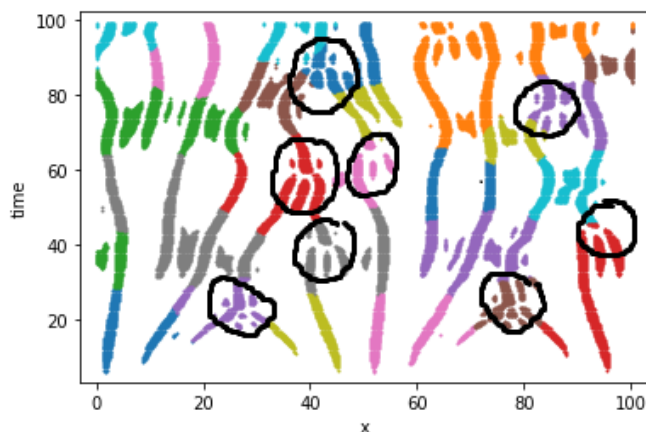


Рис. 6.4.2 Результат кластеризации пространственно-временной турбулентности, смоделированной уравнением (6.2.1.2), где черными кругами когерентные структуры

ЗАКЛЮЧЕНИЕ

В данной работе исследовалась возможность применения методов машинного обучения для анализа динамических систем. В качестве дискретной динамической системы рассматривался видеоряд, для которого необходимо предсказать последующий кадр. В качестве модели видеоряда на основе выводов о математической модели каскадов было принято решение использовать рекуррентную нейронную сеть с типом связи многие ко многим. В рамках построения рекуррентной нейронной сети рассмотрены методы построения и обучения нейронных сетей. При построении итераций каскадов тщательно разобрана задача классификации с использованием нейронных сетей. На основе результата предсказаний итераций каскада сделан вывод о возможности применения методов машинного обучения, в частности рекуррентных нейронных сетей для исследования дискретных динамических систем.

В качестве непрерывной динамической системы рассмотрена задача Коши с дифференциальным уравнением Курамото-Сивашинского четвертого порядка в частных производных. Сделан анализ параметров уравнения Курамото-Сивашинского для моделирования пространственно-временной турбулентности. Целью использования методов машинного обучения было исследования пространственно-временных когерентностей в турбулентности. Для решения данной задачи были предложены методы кластеризации. Для подбора оптимальных параметров методов кластеризации разобраны статистические методы. В результате был предложен алгоритм последовательной кластеризации пространственно-временной турбулентности с целью нахождения когерентных структур.

Исходя из результатов данной работы можно говорить, что методы машинного обучения являются пригодным инструментом для исследования динамических систем.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Указ президента Российской Федерации о развитии искусственного интеллекта в Российской Федерации №490 от 10 октября 2019 года.
2. Matthew, IRWIN Dynamic Systems Modeling / IRWIN Matthew, Wang Zheng. — Ohio: The Ohio State University, 2017. — 322 с.
3. Groshek Media, instability, and democracy: Examining the Grangercaused relationships of 122 countries from 1946 to 2003. / Groshek, J. // Journal of Communication,. — 2011. — № 61. — С. 1161-1182.
4. Busemeyer Cognitive modeling / Busemeyer, R. J, Diederich, A. — Thousand Oaks, CA: SAGE, 2009. — 228 с.
5. Wang Bridging media processing and selective exposure: A dynamic motivational model of media choices and choice response time. / Wang, Z. // Communication Research. — 2014. — № 41. — С. 1064-1087.
6. Busemeyer Dynamic systems: Mathematics / Busemeyer, R. J. — Hoboken: NJ: John Wiley & Sons, 2005. — 300 с.
7. Mark, M. M. Mathematical Modeling / M. M. Mark. — 3. — Berlin, 2003. — 312 с.
8. Kloeden, P. E. Dicrete-Time Nonautonomous Dynamical Systems / P. E. Kloeden, C. Potzsche, M. Rasmussen. — 2. — London: 2007. — 444 с.
9. Pinheiro, D. Notes on Continuous-time Dynamical Systems / D. Pinheiro. — 2. — Lisboa, Portugal: Universidade T´ecnica de Lisboa, 2011. — 39 с.
10. Вьюгин, В. В. Математические основы машинного обучения и прогнозирования / В. В. Вьюгин. — Москва: МЦНМО, 2018. — 384 с.
11. Фаустова, К. И. НЕЙРОННЫЕ СЕТИ: ПРИМЕНЕНИЕ СЕГОДНЯ И ПЕРСПЕКТИВЫ РАЗВИТИЯ / К. И. Фаустова. // Территория науки. — 2017. — № 1. — С. 1-5.
12. Горбачевская, Е. Н. КЛАССИФИКАЦИЯ НЕЙРОННЫХ СЕТЕЙ / Е. Н. Горбачевская. // Вестник Волжского университета им. В.Н. Татищева. — 2012. — № 1. — С. 1-6.
13. Michael, Nielsen Neural Networks and Deep Learning / Nielsen Michael //

Neural Networks and Deep Learning: [сайт]. — URL: <http://neuralnetworksanddeeplearning.com/> (дата обращения: 28.01.2020).

14. Tariq, Rashid Make Your Own Neural Network. / Rashid Tariq. — 1. — Ensk : CreateSpace Independent Publishing Platform, 2016. — 222 с.

15. Christopher, M. B. Pattern Recognition and Machine Learning / M. B. Christopher. — 1. — Singapore: Springer Science+Business Media, LLC, 2006. — 758 с.

16. Горбачевская, Е. Н. Learning to Generate Chairs, Tables and Cars with Convolutional Networks / Е. Н. Горбачевская. // IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. — 2017. — № 1. — С. 1-14.

17. Бендерская, Е. Н. Рекуррентная нейронная сеть как динамическая система и подходы к ее обучению / Е. Н. Бендерская, К. В. Никитин. // Научно-технические ведомости СПбГПУ. — 2013. — № 4. — С. 29-40.

18. Будыльский Д. В. GRU и LSTM: современные рекуррентные нейронные сети // Молодой ученый. — 2015. — №15. — С. 51-54.

19. Kumar, J. Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model For Cloud Datacenters / J. Kumar, R. Goomer. // Procedia Computer Science. — 2018. — № 125. — С. 676-682.

20. Андросова, Е. Е. ПРИМЕНЕНИЕ РЕКУРСИВНЫХ РЕКУРРЕНТНЫХ НЕЙРОННЫХ СЕТЕЙ / Е. Е. Андросова. // Новые информационные технологии в автоматизированных системах. — 2016. — № 1. — С. 1-8.

21. Васенков, Д. В. Методы обучение искусственных нейронных сетей / Д. В. Васенков. // Компьютерные инструменты в образовании. — 2007. — № 1. — С. 20-29.

22. Ian Goodfellow. Deep Learning / Ian Goodfellow, Yoshua Bengio, Aaron Courville // Massachusetts Institute of Technology, MIT, 2016, 781 p. URL: <http://www.deeplearningbook.org/>

23. Воронцов К. В. Машинное обучение. Видеолекции, Школа анализа данных от Яндекс URL: <https://yandexdataschool.ru/edu-process/courses/machine->

learning

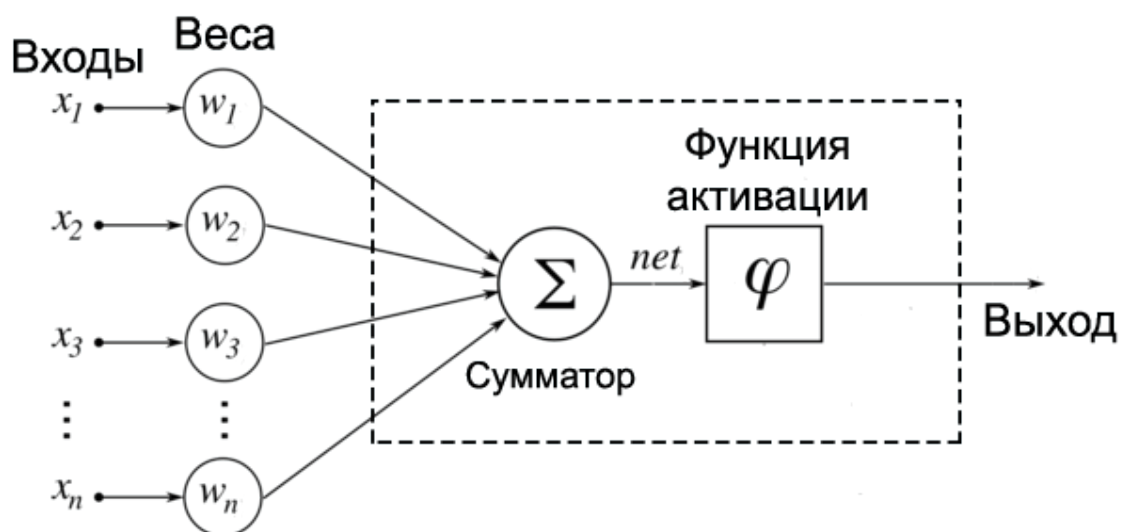
24. Aurelien Geron. Hands-On Machine Learning with Scikit-Learn and TensorFlow / Aurelien Geron // O'Reilly Media, 2017, 797 с.
25. Jake Vander Plas. Python Data Science Handbook / Jake Vander Plas // O'Reilly Media, 2017, 548 с.
26. Пройдаков, Э. М. СОВРЕМЕННОЕ СОСТОЯНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА / Э. М. Пройдаков. // Научноисследовательские исследования. — 2018. — № 1. — С. 1-25.
27. Yoshiki, Kuramoto On the Formation of Dissipative Structures in Reaction-Diffusion Systems: Reductive Perturbation Approach / Kuramoto Yoshiki, Tsuzuki Toshio. —// Progress of Theoretical Physics. — 1975. — № 54. — С. 687-699.
28. Sivashinsky, G. I. Nonlinear analysis of hydrodynamic instability in laminar flames—I. Derivation of basic equations / G. I. Sivashinsky. // Acta Astronautica. — 1977. — № 4. — С. 1177-1206.
29. Hohenberg, P. C. Chaotic behavior of an extended system / P. C. Hohenberg, B. I. Shraiman. // Physica D. — 1989. — № 37. — С. 1177-1206.
30. Leonardo, S. F. Asymptotic Dynamics of the Damped Kuramoto-Sivashinsky Equation / S. F. Leonardo. // researchgate : [сайт]. — URL: <https://www.researchgate.net/publication/334455321> (дата обращения: 04.05.2020).
31. Russell, A. E. LYAPUNOV EXPONENTS OF THE KURAMOTO-SIVASHINSKY PDE / A. E. Russell, J. E. Bunder, W. M. Trent, A. J. Roberts. // The ANZIAM. — 2019. — № 61. — С. 270-285.
32. Trefethen, L. N. Spectral Methods in MATLAB / L. N. Trefethen. — 1. — Philadelphia: SIAM, 2000. — 211 с.

ПРИЛОЖЕНИЕ А

В графическую часть выпускной квалификационной работы входят:

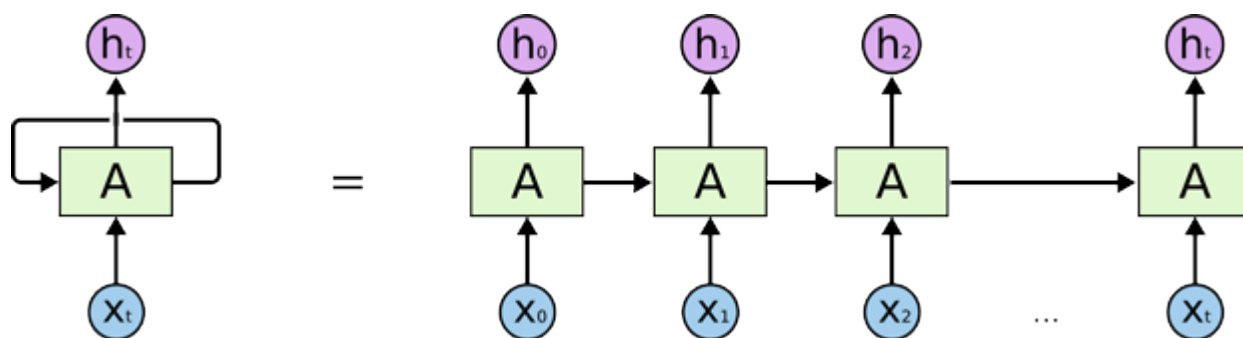
- схема простой нейронной сети;
- схема рекуррентной нейронной сети;
- файловая структура разработанного веб-приложения;
- визуализация работы с веб приложением на примере уравнения Курамото-Сивашинского;
- визуализация работы с веб приложением на примере пользовательских данных

Нейронные сети

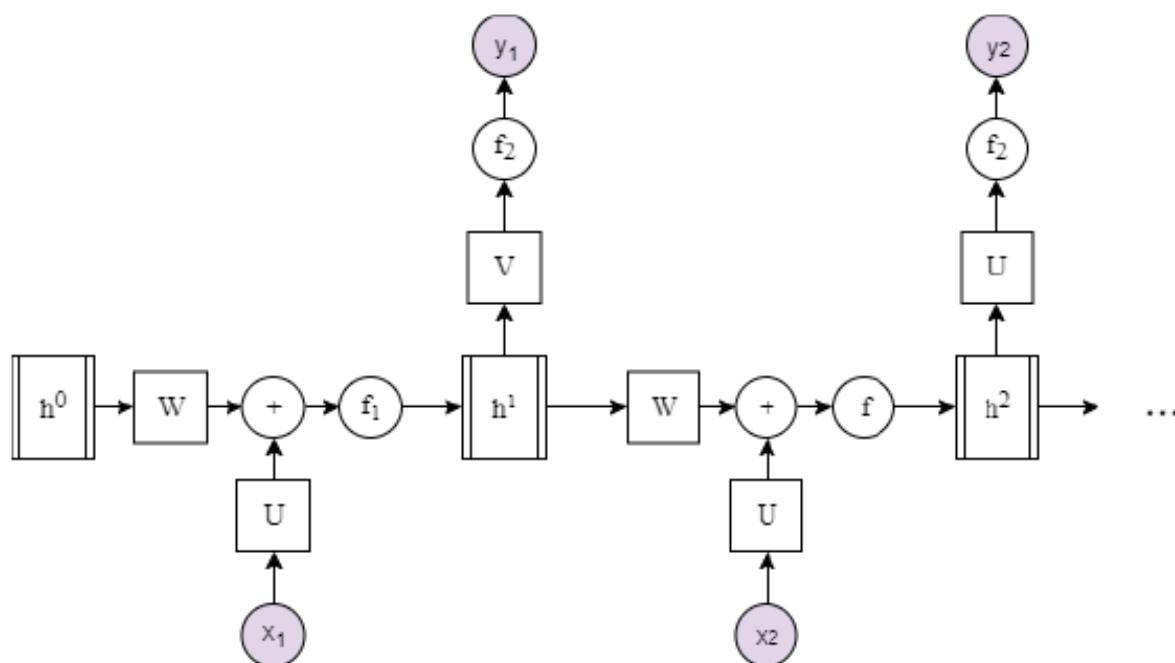


					Выпускная квалификационная работа бакалавра		
Изм	Лист	№ докум.	Подпись	Дата			
Разработ.		Максимов Д.М.					
Проверил		Соколов А.П.			Исследование итераций каскада и когерентных структур методами машинного обучения		
Зав. каф.							
Н. контр.							
Утвердил							
					Литер.	Лист	Листов
					У	1	5
					МГТУ им. Н. Э. Баумана ф-т РК гр. РК6-81		

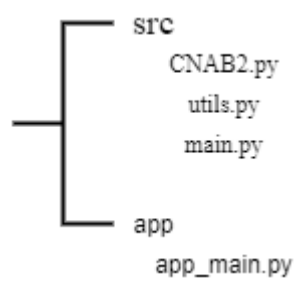
Рекуррентные нейронные сети



Рекуррентные нейронные сети с типом связи многие ко многим



Файловая структура



Интерфейс разработанного приложения включает форму вывода результата

Выберите метод кластеризации

DBSCAN

Найти оптимальный гиперпараметр eps

Введите ординату точки максимальной кривизны

0,32

-

+

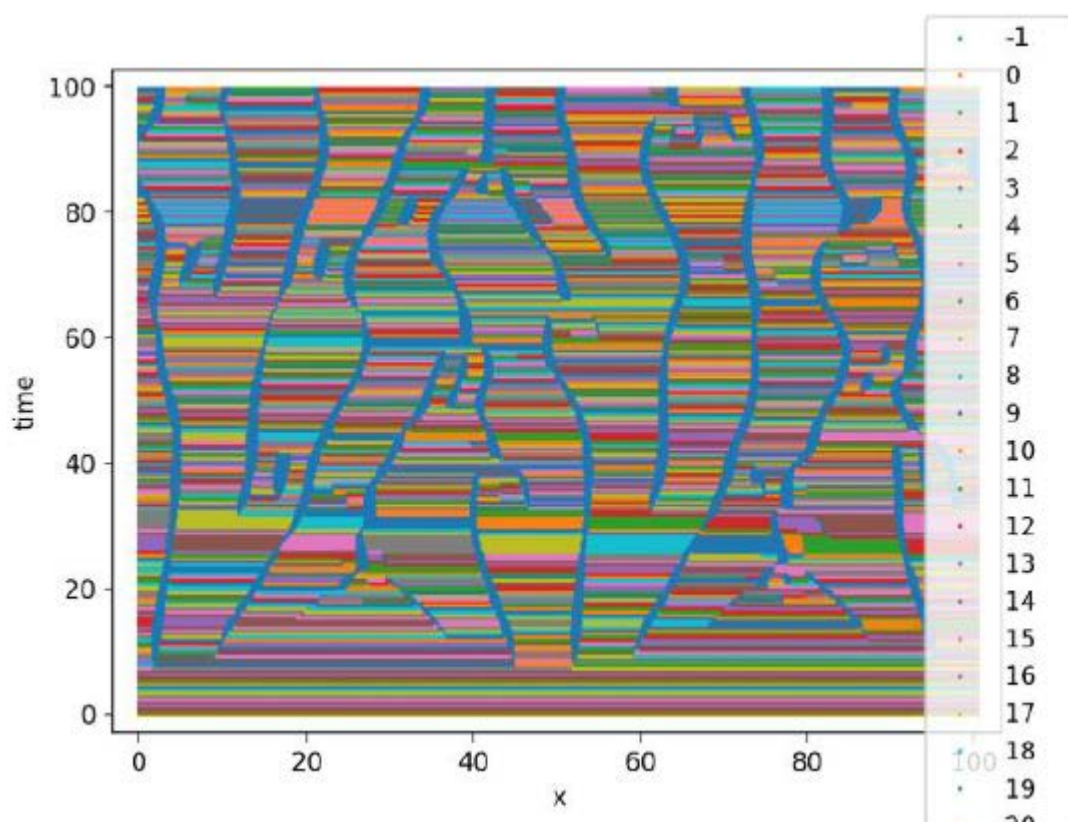
Введите количество соседей для основной точки

3

-

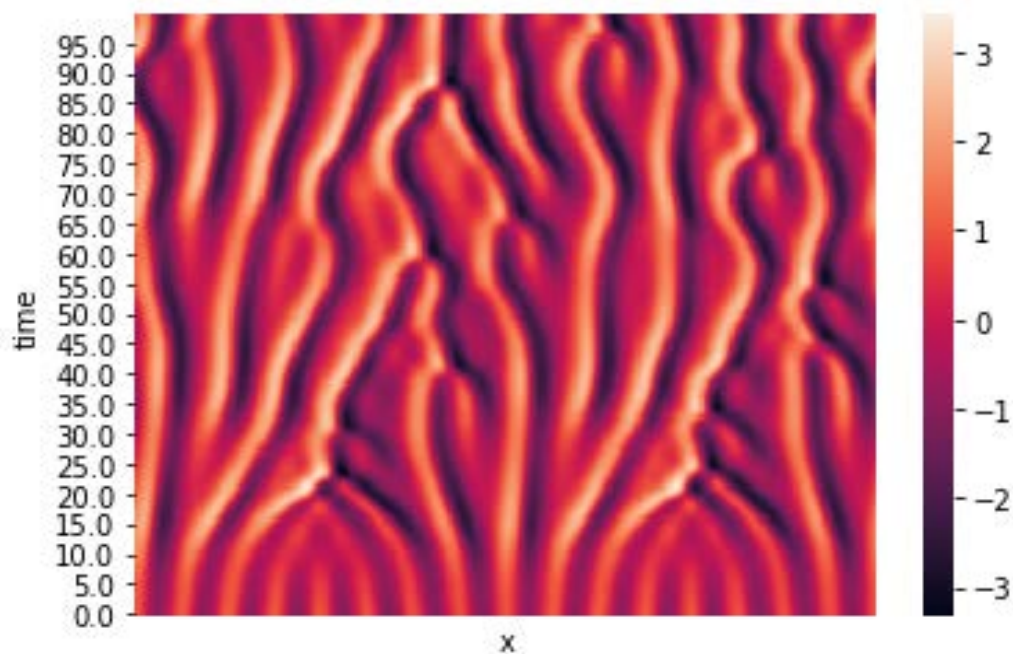
+

показать кластеризацию



Характерный результат работы приложения

Пространственно-временная турбулентность



Результат повторной кластеризации методом DBSCAN

