Министерство образования и науки Российской Федерации
Московский физико-технический институт (государственный университет)

Физтех-школа прикладной математики и информатики
Кафедра дискретной математики

Выпускная квалификационная работа магистра

Топологические методы в некоторых задачах анализа данных

Автор:
Студент 115 группы
Снопов Павел Михайлович

Научный руководитель:
д-р физ.-мат. наук, проф.
Мусин Олег Рустумович

Москва 2023

Аннотация

Топологические методы в некоторых задачах анализа данных
Снопов Павел Михайлович

Данная работа посвящена применениям устойчивых гомологий в различных задачах анализа данных. Рассмотрены задачи симплификации кривой, поиск параметра размерности вложения Такенса, а также задача анализа и изучения изменений внутренних представлений данных в нейронных сетях. Разработаны новые алгоритмы для решения задачи симплификации, а также для задачи поиска параметра размерности вложения, изучено влияние функций активации на изменение топологии внутренний представлений в нейронных сетях.

Abstract

Topological methods in some problems of data analysis

Оглавление

Глава 1

Введение

В последнее время топологические методы набирают все большую популярность в анализе данных. Область, охватывающая методы, использующие топологические идеи при решении задач анализа данных, и которые вдохновлены топологическими или алгебраическими конструкциями, называется *топологическим анализом данных*. Гланым инструментом среди топологических методов являются *устойчивые гомологии*, изначально придуманные еще более 20-ти лет назад в работах работ Герберта Эдельсбруннера[1], Гуннара Карлссона [2] и Сергея Баранникова [3]. Устойчивые гомологии позволяют изучать данные, представленные в виде облака точек, а также позволяют получить ценную топологическую информацию о данных в виде дают численных характеристик, устойчивых при малых изменениях.

Устойчивые гомологии находят свое приложение в различных областях: от анализа сложных биологических систем, как например структуры белков [4], экспрессии генов [5, 6] и нейробиологии с изучением функциональной связности мозга [7, 8], до распознавания форм [9], изучения структур сетей [10, 11], анализа изображений [12—14].

Данная работа посвящена применениям устойчивых гомологий в различных задачах анализа данных. Главной целью работы является демонстрация возможностей, которые предоставляют устойчивые гомологии в изучении сложно устроенных данных.

Данная работа устроена следующим образом:

- Глава 2 посвящена кратким напоминаниям из алгебры и топологии, и содержит в себе вводный теоретический материал по устойчивым гомологиям;

- В главе 3 решается задача симплификации кривой. Данная задача часто возникает на практике, например при генерализации картографических объектов [15]. В главе 3 разработан непараметрический алгоритм решения данной задачи на основе устойчивых гомологий;

- Глава 4 посвящена поиску наилучшего параметра размерности вложения Такенса временного ряда. Такая задача также часто встречается на практике при анализе временных рядов, и подбор подходящей размерности играет здесь ключевую роль. Параметр размерности может существенно влиять на качество моделей машинного обучения, в последствии применяемых к трансформированным данным, и также отвечает за то, насколько корректно получается восстановить подлежащую динамическую систему, которую представляет рассматриваемый временной ряд. В данной главе развиты топологические идеи, стоящие за основным методом нахождения параметра размерности –методом

ложных соседей[16] – и продемонстрирован алгоритм, использующий аппарат устойчивых гомологий, который решает данную задачу;

- В главе 5 проводится топологический анализ изменений внутренних представлений данных внутри нейронных сетей. Такой анализ демонстрирует процесс обучения нейронных сетей под новым углом и помогает лучше понять механизм обучения искусственных нейросетей. В данной главе продемонстрировано, что хорошо-обученная нейронная сеть, решающая задачу (бинарной) классификации упрощает топологию данных, изучено, как именно упрощается топология при использовании разных функций активации, в том числе при использовании модификации известного механизма skip connection в качестве функции активации.

Глава 2

Необходимые теоретические сведения из алгебраической и прикладной топологии

В этой главе мы напомним некоторые определения и факты, используемые в дальнейшем.

2.1 Симплициальные гомологии

Пусть K – симплициальный комплекс, то есть множество симплексов, такое, что

1. Для каждого симплекса из K его грани тоже лежат в K.
2. Пересечение любых двух симплексов $\sigma, \tau \in K$ либо пусто, либо является гранью и σ, и τ.

Тогда можно рассмотреть *n-мерные цепи* c, т.е. линейные комбинации с целыми коэффициентами (лишь конечное число которых ненулевые) всех ориентированных n-симплексов в K

$$c = \sum_i z_i \sigma_i^n.$$

Множество $C_n(K)$ всех n-цепей называется *n-й группой цепей* и имеет очевидную структуру свободного $\mathbb{Z}$-модуля. Прямая сумма $\bigoplus_{n\geq 0} C_n(K)$ таких групп также является свободным $\mathbb{Z}$-модулем и обозначается $C_*(K)$.

Если есть два симплициальных комплекса K и L, то говорят, что между ними существует *симплициальное отображение* $f : K \to L$, если существует отображение $f : V(K) \to V(L)$ между вершинами симплекса, такое, что если $\sigma \subseteq V(L) : \sigma \in K$, то тогда его образ $f(\sigma) \in L$. Такое отображение естественным образом индуцирует отображение между группами цепей $f_{\#,n} : C_n(K) \to C_n(L)$, а именно, оно устроено так, что образующие $C_n(K)$ – n-симплексы K – переходят в соответствующие симплексы под действием f, и такое отображение продолжается по линейности.

Из n-цепи можно получить $(n-1)$-цепь путем взятия границы. А именно, *границей* n-цепи $c = \sum_i z_i \sigma_i^n$ называется $(n-1)$-цепь

$$\partial_n(c) = \sum_{j=0}^{n} (-1)^j \sum_i \varepsilon_i \partial_j \sigma_i^n,$$

где $\partial_j \sigma_i^n = \partial_j [v_0, ..., v_n] = [v_0, ..., \hat{v_j}, ..., v_n]$ – это $(n-1)$-симплекс, порожденный всеми вершинами, кроме вершины v_j. Гомоморфизм модулей ∂_n называется *граничным оператором*. Если существует симплициальное отображение $f_{\#,n} : K \to L$, то тогда

$$f_\# \partial = \partial f_\#.$$

Лемма (Пуанкаре)**.** *Для любого $n \geq 2$ справедливо*

$$\partial_{n-1} \circ \partial_n = 0.$$

Последовательность $\mathbb{Z}$-модулей и гомоморфизмов ∂ между ними (на самом деле, вместо $\mathbb{Z}$-модулей и гомоморфизмов могут быть объекты любой аддитивной категории и морфизмы между ними [17]), для которых верно, что $\partial^2 = 0$, называется *цепным комплексом*.

Нетрудно заметить, что в таком случае $\operatorname{im} \partial_{n+1} \leq \ker \partial_n$.

Определение 1 (Группа гомологий)**.** *n-й группой гомологий $H_n(K)$ симплициального комплекса K называется $\mathbb{Z}$-модуль*

$$H_n(K) = \ker \partial_n \Big/ \operatorname{im} \partial_{n+1}.$$

2.2 Pas de Deux: категории и функторы

Конструкция групп гомологий обладает одним замечательным свойством: она функториальна. Для того, чтобы наиболее полно описать это, придется воспользоваться аппаратом теории категорией. Также он потребуется для наиболее емкого, и при этом наиболее строгого определения устойчивых гомологий. Более подробно ознакомиться с теорией категорий можно, например, в [18].

Категорией **C** называют коллекцию объектов и морфизмов между парой объектов. То есть, категория состоит из

1. (классов) объектов X,Y,Z,
2. для любых двух объектов X и Y существует класс морфизмов $\mathrm{Hom}_{\mathbf{C}}(X,Y)$ (если категория понятна из контекста, то индекс будем опускать),
3. для любого объекта X существует единичный морфизм $1_X \in \mathrm{Hom}(X,X)$,
4. на морфизмах задана операция композиции: для любого пары морфизмов $f \in \mathrm{Hom}(X,Y)$ и $g \in \mathrm{Hom}(Y,Z)$ существует морфизм $gf \in \mathrm{Hom}(X,Z)$,
5. для любого морфизма $f \in \mathrm{Hom}(X,Y)$ верно, что $1_Y f = f = f 1_X$,
6. операция композиции ассоциативна.

Можно рассматривать отображения между категориями. *(Ковариантным) функтором* $F : \mathbf{C} \to \mathbf{D}$ между категориями **C** и **D** называют такое отображение, что

1. $\forall c \in \mathbf{C}, F(c) \in \mathbf{D}$
2. $\forall f : c \to c', F(f) : Fc \to Fc' \in \mathbf{D}$
3. $\forall c \in \mathbf{C}, F(1_c) = 1_{F(c)}$
4. $\forall f,g \in \mathbf{C}, F(fg) = F(f)F(g)$

Последние два свойства являются определяющими для функтора и называются *условиями функториальности.*

Вернемся к группам гомологий. Если имеется два симплициальных комплекса K и L, и симплициальное отображение f между ними, то оно индуцирует гомоморфизм модулей

$$f_{*,n} : H_n(K) \to H_n(L).$$

А именно, так как существует отображение между группами цепей $f_{\#,n}$, то

Более того, если существует цепочка симплициальных отображений

$$K \xrightarrow{f} L \xrightarrow{g} M,$$

то она индуцирует цепочку отображений в гомологиях:

$$H_n(K) \xrightarrow{f_{*,n}} H_n(L) \xrightarrow{g_{*,n}} H_n(M).$$

Таким образом, группа n-ых гомологий является функтором из категории симплициальных комплексов в категорию $\mathbb{Z}$-модулей/абелевых групп:

$$H_n : \mathbf{Simp} \to \mathbb{Z}-\mathbf{Mod}$$

2.3 Персистентные гомологии

Под облаком точек D здесь и далее мы будем подразумевать конечное множество в $\mathbb{R}^n$. О персистентных (устойчивых) гомологиях можно думать как об адаптации понятия гомологии к облаку точек.

А именно, имея облако точек, можно построить семейство симплициальных комплексов, параметризованное некоторым частично упорядоченным множеством (например, на практике зачастую это $\mathbb{R}$ или его подмножества). Такая параметризация естественным образом задает отображения между получаемыми симплициальными комплексами. Тогда устойчивыми гомологиями будут в точности гомологии полученных комплексов вместе с индуцированными гомоморфизмами между ними.

Определим теперь устойчивые гомологии формально, следуя [19].

Зафиксируем некоторое частично упорядоченное множество $T \subseteq \mathbb{R}$ и соответствующую ему категорию $\mathbf{T}$. Тогда функтор

$$F : \mathbf{T} \to \mathbf{Simp}$$

называется *фильтрацией.* Фильтрация является частным случаем более общего понятия представления частично упорядоченного множества. А именно, *представлением частично упорядоченного множества* называют

Наиболее важным для нас примером фильтрации на облаке точек D будет являться *фильтрация Вьеториса-Рипса*

$$\mathrm{Rips}(D) : [\mathbf{0},\infty) \to \mathbf{Simp} : r \mapsto X(\mathcal{N}(D)_r),$$

где $\mathcal{N}(D)_r$ – это граф соседей радиуса r, т.е. такой граф, вершины которого –это исходное облако точек D, и ребро существует, если расстояние между двумя точками не превосходит r. $X(G)$ – это кликовый комплекс графа G, т.е. наибольший (по включению) симплициальный комплекс, чей 1-скелет – это G.

Алгоритм 1: Алгоритм построения комплекса Вьеториса—Рипса

Исходные параметры: Облако точек X, вещественное число $\alpha > 0$.
Результат: Симплициальный комплекс Вьеториса—Рипса
Для каждой точки x строим её α-окрестность $B_\alpha(x)$;
$i = 1$;
до тех пор, пока $i+1$ *окрестностей попарно имеют непустое пересечение* **выполнять**
  строим i-ый симплекс на соответствующих вершинах;
  $i \leftarrow i + 1$;
конец

Композиция функтора гомологий $H_n : \mathbf{Simp} \to \mathbb{Z}-\mathbf{Mod}$ с фильтрацией даст функтор

$$M_n : \mathbf{T} \to \mathbb{Z}-\mathbf{Mod}.$$

Такие функторы называются *модулями устойчивости*. Совокупность же таких модулей $M_n : \mathbf{T} \to \mathbb{Z}-\mathbf{Mod}$, варьирующаяся по всем размерностям, называется *устойчивыми гомологиями* $PH_*(T)$ фильтрации T.

На практике часто используют фильтрацию Вьеториса-Рипса, тогда весь метод выглядит так: имея облако точек D, варьируя радиус r, строится возрастающая последовательность симплициальных комплексов

$$K_0 \subseteq K_1 \subseteq ... \subseteq K_s \subseteq ...,$$

где каждый симплициальный комплекс K_j является кликовым комплексом графа $\mathcal{N}(D)_j$ соседей радиуса j. Применяя к полученной последовательности функтор гомологий, получается последовательность $\mathbb{Z}$-модулей

$$H_n(K_0) \to H_n(K_1) \to ... \to H_n(K_s) \to ....$$

Это и есть устойчивые гомологии.

Из такого определения сразу видно, что устойчивые модули – это просто представление частично упорядоченного множества $(T, \leq)$. В хороших случаях имеется целая плеяда утверждений, которые описывают структуру таких модулей. Ниже представлено одно из наиболее общих, и одновременно наиболее подходящих для наших задач, утверждений.

Теорема 1 (Crawley-Boevey[20])**.** *Пусть k – поле, и $T \subseteq \mathbb{R}$. Тогда любое поточечно-конечное представление V частично упорядоченного множества $(T, \leq)$ над k имеет вид*

$$V \simeq \bigoplus_{I \in \mathcal{B}(V)} k_I,$$

где I – некоторый интервал $[b,d]$ в T, а k_I – это представление, которое имеет вид

$$... \to \underset{b-1}{0} \to \underset{b}{k} \xrightarrow{\mathrm{id}} ... \to \underset{d}{k} \xrightarrow{\mathrm{id}} \underset{d+1}{0} \to ...$$

Множество $\mathcal{B}(V)$ называется *баркодом* данного устойчивого модуля. Каждый интервал в баркоде содержит в себе информацию о том, когда определенный топологический признак появился – время рождения, и когда исчез – время смерти.

Эту же информацию можно представлять иначе: каждый интервал $[b,d]$ баркода можно рассматривать как точку с координатами (b,d) на (расширенной) плоскости. Так как $0 < b < d$, то эти точки всегда находятся в положительном квадранте выше диагонали $y = x$. Такое представление баркода называется *диаграммой устойчивости*.

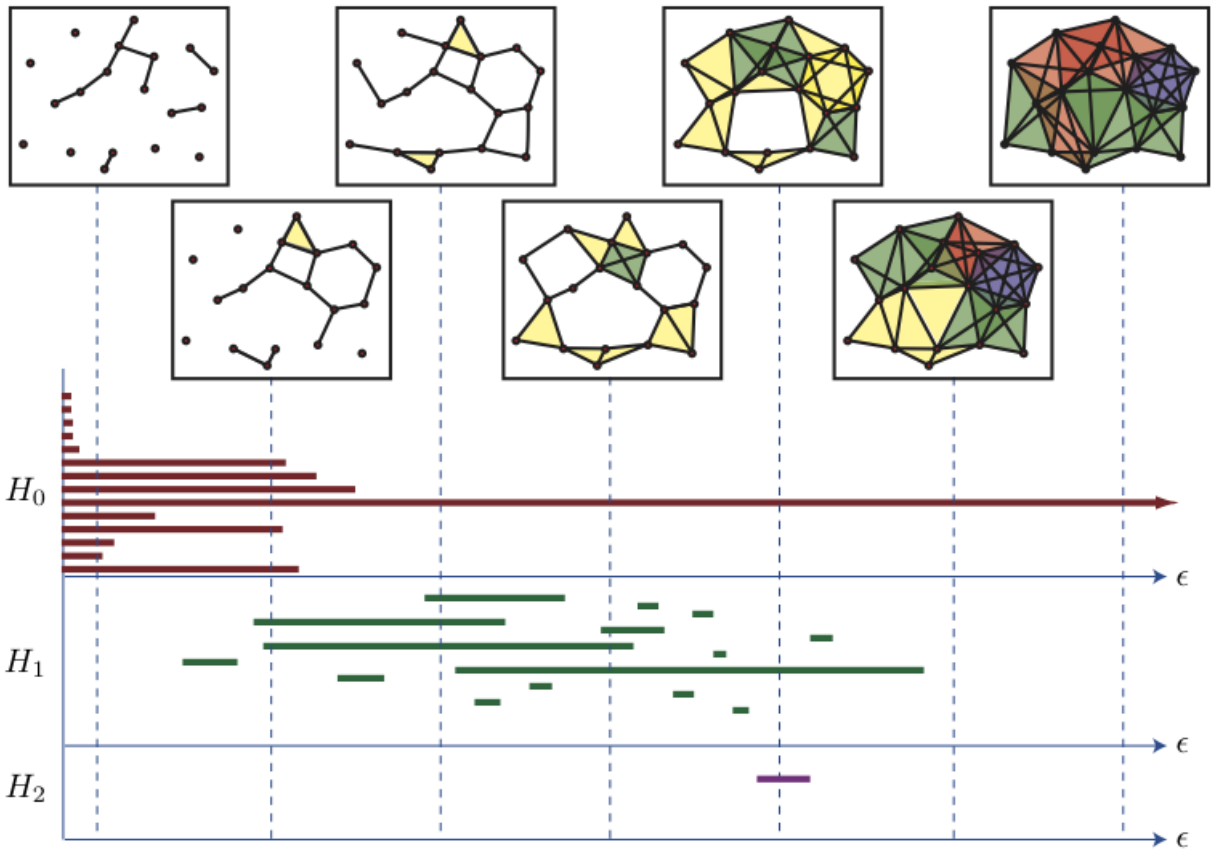


Рис. 2.1: Пример фильтрации и полученного по ней баркода

2.3.1 Стабильность диаграмм устойчивости

Одним из ключевых утверждений является теорема об изометрии. Она утверждает, грубо говоря, что если есть некоторое облако точек D, то его небольшие шевеления $D + \varepsilon$ не сильно повлияют на диаграмму устойчивости $PD(X)$. Для того, чтобы привести строгую формулировку теоремы, нужно проделать некоторую работу.

Пусть $\mathcal{D}$ – множество устойчивых диаграмм. На нем можно ввести (естественную) метрику:

$$W_p(B, B^{'}) = \inf_{\gamma: B \to B^{'}} \big(\sum_{u \in B} u - \gamma(u)_{\infty}^{p} \big)^{\frac{1}{p}},$$

где $1 \leq p < \infty$, $B, B^{'}$ – диаграммы персистентности. Такую метрику называют *$p-$метрику Васерштейна*. Другой естественной метрикой является т.н. *bottleneck distance* W_{∞}:

$$W_{\infty}(B, B^{'}) = \inf_{\gamma: B \to B^{'}} \sup_{u \in B} u - \gamma(u)_{\infty},$$

где γ – биекция между двумя диаграммами. Таким образом, $\mathcal{D}$ с любой из указанных выше метрик образует метрическое пространство.

Можно немного изменить определение устойчивых диаграмм – пусть теперь у нас всегда она содержит диагональ $\Delta = \{(x,y) \in \mathbb{R}^2 | x = y\}$. Для таких диаграмм p-метрика Васерштейна обобщается естественным способом. Пустую диаграмму, содержащую только диагональ, будем обозначать через $B_{\emptyset}$. Тогда пространством устойчивых диаграмм можно считать следующее пространство:

$$\mathcal{D} = \{B | W_p(B, B_{\emptyset}) < \infty\}.$$

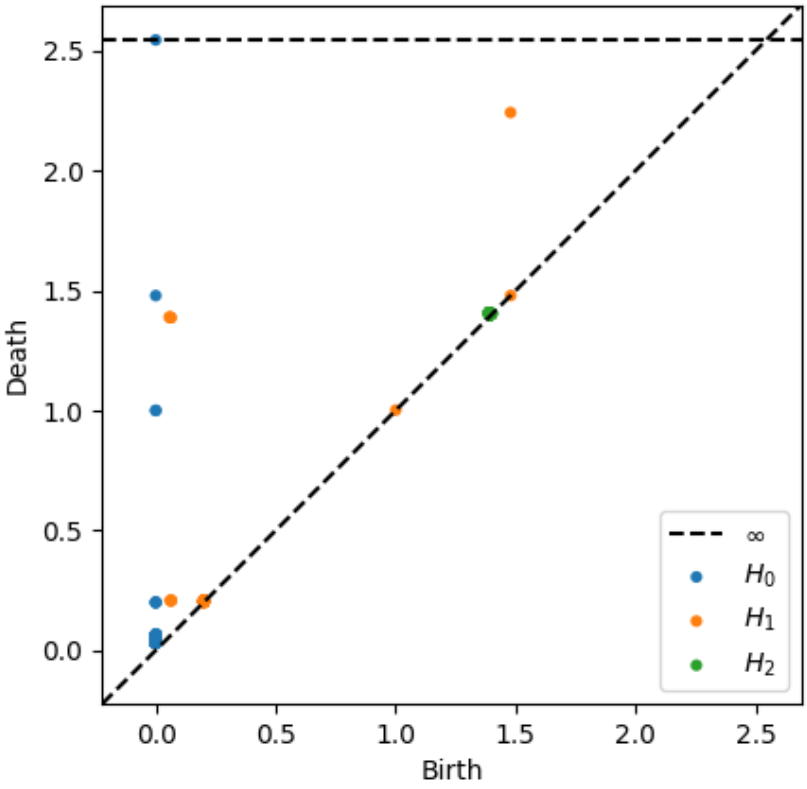


Рис. 2.2: Пример диаграммы устойчивости

Но можно вводить метрики непосредственно между устойчивыми модулями. А именно, пусть V,W – два модуля устойчивости над $\mathbb{R}$ и $\varepsilon \geq 0$. Назовем ε-*чередованием* между V и W два семейства морфизмов $(\phi_i : V_i \to W_{i+\varepsilon})_{i\in\mathbb{R}}$ и $(\psi_i : W_i \to V_{i+\varepsilon})_{i\in\mathbb{R}}$, таких, что диаграммы 2.3 коммутируют для любого $i \leq j \in \mathbb{R}$.

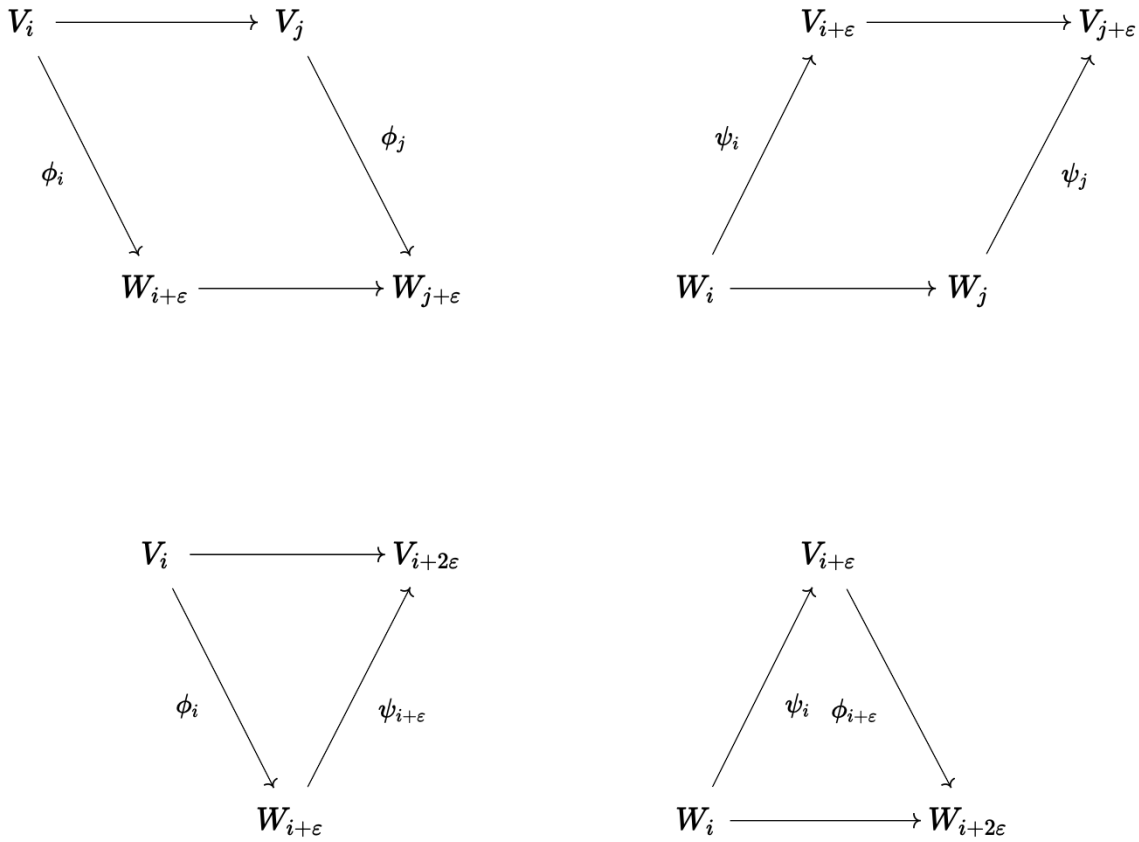


Рис. 2.3: Коммутативные диаграммы для ε-чередования

Заметим, что ε-чередование можно определить, используя чисто теоретико-категорные конструкции. А именно, назовем семейство морфизмов $(\phi_i : V_i \to W_{i+\varepsilon})_{i\in\mathbb{R}}$, таких, что диграммы 2.4 коммутируют для всех пар $i \leq j \in \mathbb{R}$, *морфизмом степени* ε между V и W.

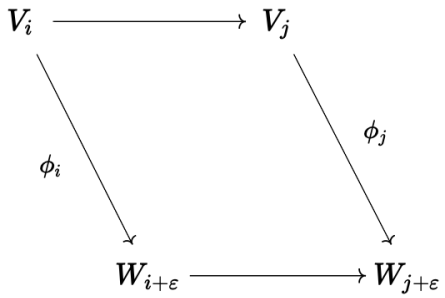


Рис. 2.4: Коммутативные диаграммы для морфизма степени ε

Пусть

$$\mathrm{Hom}^{\varepsilon}(V,W) \coloneqq \{\text{морфизмы } V \to W \text{ степени } \varepsilon\}$$
$$\mathrm{End}^{\varepsilon}(V) \coloneqq \{\text{морфизмы } V \to W \text{ степени } \varepsilon\}$$

Тогда композиция двух таких морфизмов дает отображение

$$\mathrm{Hom}^{\varepsilon'}(V,W) \times \mathrm{Hom}^{\varepsilon}(U,V) \to \mathrm{Hom}^{\varepsilon+\varepsilon'}(U,W).$$

Если $U = W$, тогда существует *морфизм с сдвигом на* $1_U^{(\varepsilon+\varepsilon')} \in \mathrm{End}^{(\varepsilon+\varepsilon')}(U)$. Этот морфизм "перемещает"U на $\varepsilon + \varepsilon'$. Тогда два модуля устойчивости V и W над $\mathbb{R}$ является ε-*чередуемыми*, если существует $\phi \in \mathrm{Hom}^{\varepsilon}(V,W)$ и $\psi \in \mathrm{Hom}^{\varepsilon}(W,V)$, такие, что

$$\psi \circ \phi = 1_V^{2\varepsilon} \text{ и } \phi \circ \psi = 1_W^{2\varepsilon}$$

Тогда можно определить *расстояние чередования* $d_i(V,W)$ между V и W следующим образом:

$$d_i(V,W) \coloneqq \inf\{\varepsilon \geq 0 : \text{существует } \varepsilon\text{-существование между } V \text{ и } W\}.$$

Теорема об изометрии звучит следующим образом:

Теорема 2 (Теорема об изометрии [21])**.** *Пусть V,W – два модуля устойчивости над $\mathbb{R}$. Тогда*

$$W_\infty(PD(V), PD(W)) = d_i(V,W).$$

2.3.2 Как вычисляются устойчивые гомологии на практике

Вычислительная сторона устойчивых гомологий за последние несколько лет проделала огромный путь, и сейчас существуют несколько пакетов, в которых эффективно вычисляются устойчивые гомологии (например, `Ripser` на `C++` [22], его порт на `Python` [23], пакет `GUDHI` на `C++` и `Python` [24] или пакет `Ripserer` на языке `Julia` [25]).

Ниже изложим первый из предложенных алгоритмов, по совместительству являющийся и наиболее простым для объяснения, и сводящимся к обычному методу Гаусса. Дальнешее изложение повторяет изложение в [26].

Пусть дана стабилизирующаяся фильтрация $\{K_t\}$, т.е. существует такой T, что $\forall t > T : K_t = K_T = K$. Также будем считать, что фильтрация K_t подробная, т.е. на каждом шаге фильтрации приклеивается ровно один симплекс. Это достаточно удобно, т.к. позволяет лучше отслеживать происходящее на каждом шаге. Все гомологии будем вычислять разом над $\mathbb{R}$. Для этого сформируем векторное пространство $C_*(K) \coloneqq \oplus_i C_i(K,\mathbb{R})$ с базисом из всех непустых симплексов K, обозначим их число за N. Тогда взятие границы – симплициальный дифференциал ∂ – будем линейным отображением $C_*(K) \to C_*(K)$.

Сформируем матрицу D оператора ∂ в данном базисе, при этом считая, что симплексы упорядочены согласно фильтрации K_t. В этой матрице элемент a_{ij} ненулевой только если i-ый симплекс является подсимплексом j-симплекса, и его размерность на 1 меньше размерности j-симплекса.

В j-ом столбце найдем самый нижний ненулевой элемент и обозначим его за $\mathrm{low}(j)$. Получаем отображение $\mathrm{low} : [N] \to [N] \cup \{0\}$. Будем считать, что такая матрица имеет *приведенный вид*, если отображение low инъективно на ненулевых столбцах, т.е. в ней нет двух столбцов, у которых нижние элементы расположены на одной строке. Приведенную матрицу можно получить из метода Гаусса, примененного к столбцам.

Пусть теперь матрица M – приведенная вид матрицы D. Пусть $\mathrm{Zero}_i(M)$ – число ненулевых столбцов матрицы M, соответствующих i-мерным симплексам, а $\mathrm{Low}_i(M)$

– число нижних ненулевых элементов в ненулевых строках матрицы, находящиеся в строчках, соответствующих i-симплексам, т.е.

$$\mathrm{Low}_i(M) \coloneqq \#\{j \in \mathrm{low}([N]) : \dim I_j = i\}.$$

Тогда

$$\beta_i(K) = \mathrm{Zero}_i(K) - \mathrm{Low}_i(K).$$

Скажем теперь, что два симплекса I_j, I_s, таких, что $j < s$, образуют *сопряженную пару*, если

- $\dim I_s = \dim I_s + 1$,
- Симплекс I_j размерности i *положительный*, т.е. при его добавлении число Бетти β_i увеличилось, а симплекс I_s *отрицательный*, т.е. число Бетти β_{i-1} уменьшилось на 1,
- Гомология, рожденная в момент времени j умирает в момент времени s.

Сопряженные пары можно вычислять из матрицы M, а именно, если $j = \mathrm{low}(s) \neq 0$ в M, то I_j является положительным симплексом, I_s является отрицательным, и (I_j, I_s) образуют сопряженную пару. Если при этом $\mathrm{low}() = 0$ и $r \notin \mathrm{low}([N])$, то I_r является положительным симплексом, который ни с чем не спарился.

В итоге, задача вычисления устойчивых гомологий оказалась не сложнее задачи вычисления обычных гомологий.

Глава 3

Разработка алгоритма симплификации кривой на основе устойчивых гомологий

3.1 Введение

Симплификация кривой $\mathbb{R} \to \mathbb{R}$, аппроксимированной большим количеством точек – часто встречающаяся на практике задача. Например, это задача играет огромную роль в вопросе генерализации картографических объектов [15]. Основной сложностью зачастую является выбор параметра упрощения. Несмотря на это, во многих задачах наличие такого гиперпараметра является ключевым, и сам метод упрощения, обладая таким параметром, становится более гибким.

В данной работе будет разработан алгоритм симплификации на основе аппарата устойчивых гомологий, который при этом не подразумевает выбор параметров, однако разработанный алгоритм также можно модифицировать, добавив такой параметр упрощения.

Похожий алгоритм был предложен в [27], который был изучен в [28] в контексте качества сглаживания. В нашей же работе алгоритм упрощения на основе устойчивых гомологий будет изучен с точки зрения упрощения исходной кривой с минимальной потерей в качестве.

3.2 Алгоритм Рамера-Дугласа-Пекера

Для начала рассмотрим существующие подходы к решению такой задачи. Стандартным алгоритмом для задачи симплификации является алгоритм Рамера-Дугласа-Пекера. Суть алгоритма состоит в том, чтобы по данной ломаной, аппроксимирующей кривую, строится ломаная с меньшим числом точек. Данная упрощенная ломаная состоит из подмножества точек исходной ломаной. У этого алгоритма есть один параметр – ширина коридора или полосы допуска ε, которая зависит от масштаба. На основе этого значения принимается решение о добавлении точки исходной ломаной в множество точек упрощенной.

Исходно алгоритм принимает на вход все точки, аппроксимирующие кривую. Далее берется первая x_0 и последняя точки списка x_{-1}, и для каждой другой точки рассчитывается расстояние до прямой, проходящей через выбранные точки. Если для некоторой точки x^* расстояние до прямой больше ε, то такая точка сохраняется в множество точек упрощенной ломанной, и алгоритм рекурсивно запускается на от-

резках $[x_0, x^*]$ и $[x^*, x_{-1}]$. Если же таких точек не нашлось, то все «значимые» точки уже выбраны алгоритмом, и тогда получившееся множество точек будет являться множеством точек, дающее упрощенную ломанную. Псевдокод такого алгоритма приведен в 2.

Алгоритм 2: Алгоритм Рамера-Дугласа-Пекера

Исходные параметры: Список точек X, вещественное число $\varepsilon > 0$.
Результат: Список X', содержащий меньшее число точек
`RDP(`X,ε`)`:
  $d_{max} \leftarrow \max_j \{\text{dist}(X_j, \text{line}(X[0], X[-1]))\}$;
  если $d_{max} > \varepsilon$ **тогда**
    $X' \leftarrow$`RDP(`acb`)`
  конец
  иначе
    $X' \leftarrow \left[X[0], X[-1]\right]$
  конец
  возвратить X'

Алгоритм Рамера-Дугласа-Пекера сохраняет минимумы и максимумы. В целом любой алгоритм симплификации должен сохранять экстремальные точки. Такого же эффекта – сохранения экстремумов – можно добиться и при помощи устойчивых гомологий, используя правильную фильтрацию.

Причем, используя устойчивые гомологии, можно избавиться от гиперпараметра, который присутствует в алгоритме Рамера-Дугласа-Пекера и затрудняет его использование.

3.3 Симплификация на основе устойчивых гомологий

Имея набор точек X, аппроксимирующие некоторую кривую, получаемую кривую можно рассматривать как 1-сиплициальный комплекс, где 1-симплексами будут отрезки, соединяющие соседние точки набора. Тогда можно рассмотреть следующую фильтрацию

$$F : \mathbf{R} \to \mathbf{Simp} : r \mapsto X^{-1}(-\infty, r).$$

То есть элементом фильтрации является 1-симплициальный комплекс, вершины x_i которого меньше r.

По данной фильтрации можно посчитать устойчивые гомологии. В силу конструкции симплициального комплекса, только 0-мерные устойчивые гомологии будут нетривиальны. Более того, точки экстремума такого отображения будут отвечать за рождение и смерть 0-мерных циклов (что следует общему механизму, развитому в теории Морса, см. [29]). Полученные в результате подсчета точки на диаграмме устойчивости будут кодировать в себе как раз такие экстремальные точки. В качестве симплифицированного временного ряда можно рассмотреть все точки исходного набора точек, которые либо отвечали за рождение, либо за смерть 0-мерных циклов. Назовем такой алгоритм симплификации алгоритмов симплификации на основе устойчивых гомологий или `TopoSimplification`.

Алгоритм 3: Алгоритм `TopoSimplification`

Исходные параметры: Список точек X
Результат: Список X', содержащий меньшее число точек
Построить фильтрацию $F : r \mapsto X^{-1}(-\infty, r)$;
Посчитать 0-устойчивые гомологии по полученной фильтрации F;
$X' \leftarrow$ {те точки X, которые соответствуют рождению/смерти 0-цикла на диаграмме устойчивости} $\cup$ {первая и последняя точка списка};
возвратить X';

Предложенный алгоритм теперь не обладает недостатком алгоритма Рамера-Дугласа-Пекера, он является непараметрическим. При этом он способен кардинально уменьшить число точек, почти не потеряв в качестве приближения. Однако в некоторых задачах может потребоваться параметризованная версия алгоритма. Тогда можно ввести параметр $\varepsilon \geq 0$, отвечающий за расстояние до диагонали на диаграмме устойчивости. Если расстояние точки на диаграмме меньше ε, то такая точка исключается из построения симплифицированного временного ряда.

Алгоритм 4: Алгоритм `TopoSimplification` с параметром

Исходные параметры: Список точек X, параметр $\varepsilon \geq 0$
Результат: Список X', содержащий меньшее число точек
Построить фильтрацию $F : r \mapsto X^{-1}(-\infty, r)$;
Посчитать 0-устойчивые гомологии по полученной фильтрации F;
Исключить из диаграммы те точки, расстояние до диагонали от которых меньше ε.
$X' \leftarrow$ {те точки X, которые соответствуют рождению/смерти 0-цикла на новой диаграмме устойчивости} $\cup$ {первая и последняя точка списка};
возвратить X';

Пример работы такого алгоритма представлен на рис. **??**. На нем изображена диаграмма устойчивости с 0-мерными устойчивыми гомологиями, посчитанными по датасету SPY (синие точки) с указанной выше фильтрацией. Оранжевым выделены точки, чье расстояние до диагонали больше $\varepsilon = 0.1$. Такие точки на диаграмме порождены точками датасета, которые и войдут в упрощенный датасет.

3.4 Сравнение двух методов

Сравним алгоритм симплификации на основе устойчивых гомологий с алгоритмом Рамера-Дугласа-Пекера, используя ℓ^2 метрику. Нас будет интересовать, какое количество точек получается, применяя первый или второй метод, и какие значения метрики полученные упрощенные ломанные достигают.

3.4.1 Описание данных

Сравнение алгоритмов будет проведено на 3 временных рядах ежедневных цен закрытия акций QQQ (2517 точек) и SPY (2517 точек) на протяжении 10 лет с 2012 по 2022 год, а также 7-летняя история цены биткоина BTC в USD (2664 точек). Эти данные были получены с помощью Yahoo Finance [30].

Для корректного сравнения результатов работы алгоритмов потребуется также провести нормировку данных. Для этого приведем все данные к одному масштабу, так, чтобы все значения лежали в отрезке [0,1]. Сделать это можно при помощи

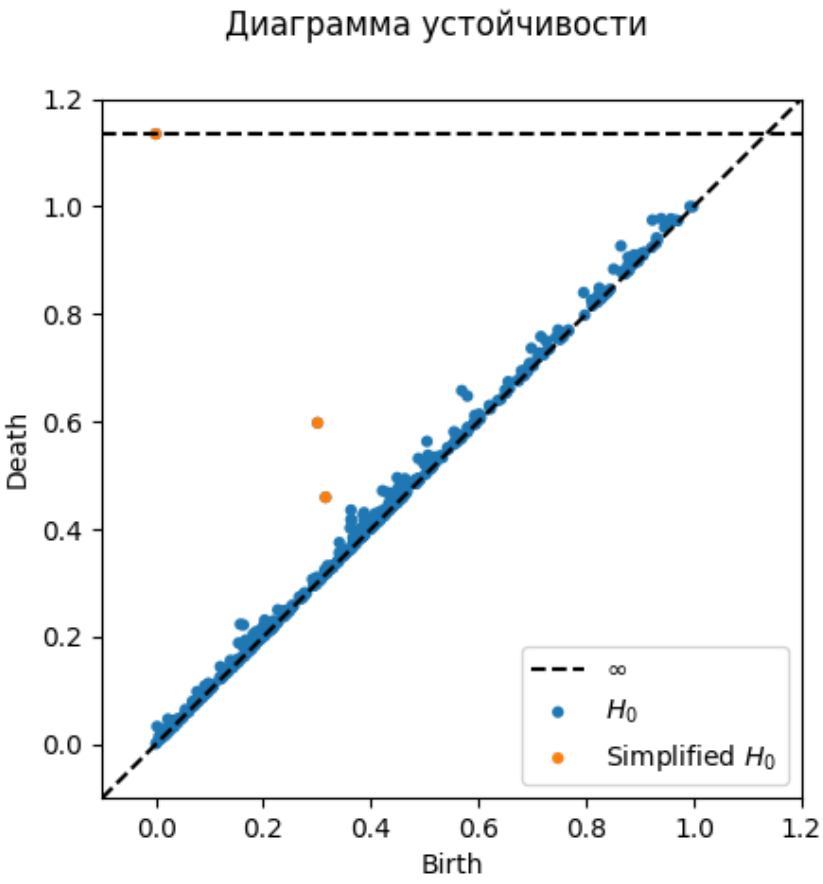


Рис. 3.1: Пример работы алгоритма `TopoSimplification` с параметром на датасете SPY с параметром $\varepsilon = 0.1$

Рис. 3.2: Временной ряд QQQ

Рис. 3.3: Временной ряд SPY

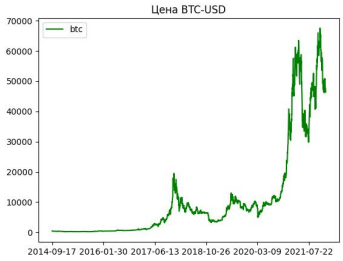


Рис. 3.4: Временной ряд BTC

следующей нормализации:

$$\text{data} := \frac{\text{data} - \text{data.min}()}{\text{data.max}() - \text{data.min}()}$$

3.4.2 Описание методы сравнения

Как было сказано выше, будем сравнивать алгоритмы симплификации с исходной кривой, считая ℓ^2 расстояние между кривой и ее упрощенной версией. При этом, изменяя гиперпараметры алгоритмов, будем получать разные варианты упрощенных кривых.

Так как конечной целью алгоритма симплификации является непосредственно упрощение кривой, будем также отслеживать, сколько точек упрощенная кривая в себе содержит. Исходя из этого, будем считать, что какой-то из алгоритмов справляется с задачей лучше, если либо при одинаковом числе точек отклонение результата работы этого алгоритма от исходной кривой меньше, либо при одинаковом отклонении от исходной кривой, упрощенная лучшим алгоритмом кривая содержит в себе меньшее число точек.

3.4.3 Результаты

Результаты сравнений представлены в таблице 3.1 и на рис. 3.6. В таблице видно, что алгоритм Рамера-Дугласа-Пекера при одинаковом числе точек достигает меньшей ошибки по сравнению с алгоритмом `TopoSimplification`.

Несмотря на это, сам алгоритм показал себя неплохо: так, рассматривая непараметрическую версию, алгоритм уменьшает примерно в 2 раза число точек (число точек в первом столбце таблицы 3.1). Помимо этого, алгоритм все еще доставляет небольшую ошибку при упрощении, см. рис. 3.6. Для сравнения, на рис. 3.7 продемонстрировано упрощение с помощью алгоритма Рамера-Дугласа-Пекера.

3.5 Заключение

Таким образом, на основе устойчивых гомологий был разработан непараметрический алгоритм симплификации кривой, а также его модификация с параметром упрощения. Данный алгоритм на рассмотренных данных показал себя хуже в смысле метрики ℓ^2, чем алгоритм Рамера-Дугласа-Пекера. Несмотря на это, непараметрическая версия алгоритма способна сильно уменьшать число данных, упрощая исходный датасет, не сильно теряя в качестве. Более того, алгоритм на основе устойчивых гомо-

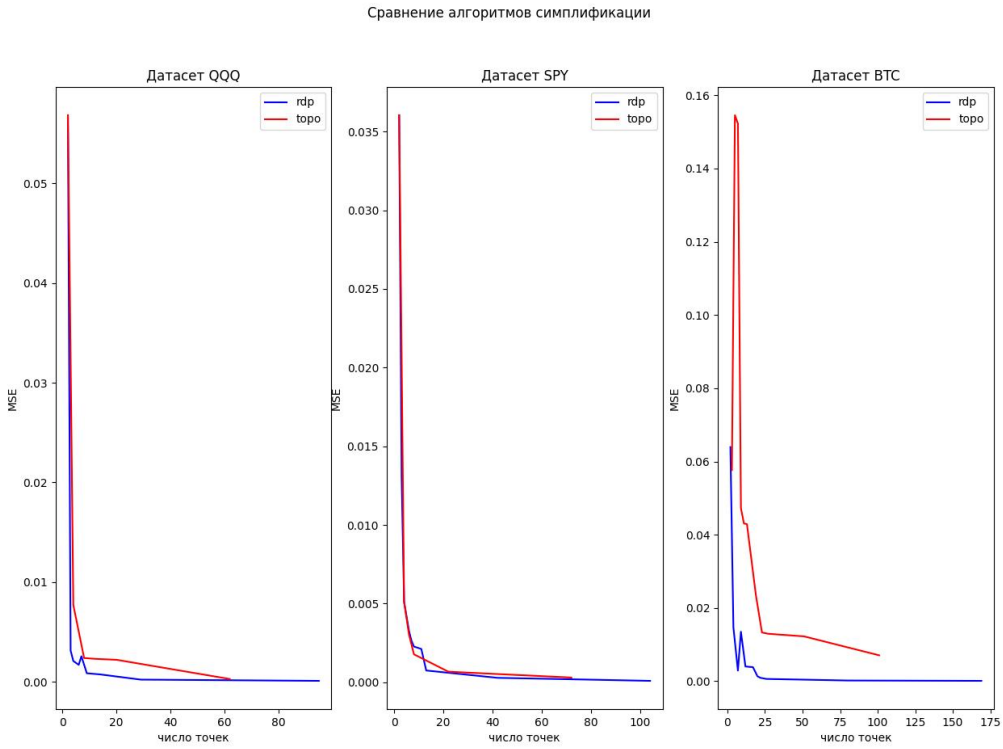


Рис. 3.5: Результаты сравнения алгоритмов симплификации

Рис. 3.6: Сравнение временных рядов с их упрощенными вариантами (упрощение с помощью TopoSimplification)

Рис. 3.7: Сравнение временных рядов с их упрощенными вариантами (упрощение с помощью алгоритма Рамера-Дугласа-Пекера)

Таблица 3.1: Результаты сравнения алгоритмов симплификации

(a) QQQ

	1240 точек	555 точек	410 точек	230 точек	62 точек
Алгоритм РДП	4.9601e-07	4.6714e-06	8.4759e-06	2.2759e-05	1.8783e-04
TopoSimplification	7.4128e-06	1.5537e-05	2.2260e-05	3.6137e-05	2.7252e-04

(b) SPY

	1300 точек	730 точек	500 точек	230 точек	72 точки
Алгоритм РДП	4.6440e-07	3.4835e-06	6.6777e-06	3.3361e-05	1.3828e-04
TopoSimplification	7.5892e-06	1.5489e-05	2.3995e-05	7.2278e-05	3.0386e-04

(c) BTC

	1350 точек	700 точек	462 точки	222 точки	83 точки
Алгоритм РДП	8.1524e-08	1.6649e-06	1.2952e-05	6.4153e-05	1.8939e-04
TopoSimplification	2.1365e-05	2.7184e-05	3.9247e-05	1.8895e-04	1.0836e-02

логий прямым образом обобщается на объекты старших размерностей, что выгодно его отличает от алгоритма Рамера-Дугласа-Пекера.

Глава 4

Поиск параметра размерности вложения Такенса с помощью устойчивых гомологий

4.1 Введение

Изучение временных рядов является одной из основных задач анализа данных. Один из подходов к изучению и анализу временных рядов базируется на теории динамических систем, т.н. *нелинейный анализ временных рядов*. Такой подход предполагает, что временной ряд является детерминированным, существует некоторая (дискретная) динамическая система (M, ϕ), и искомый временной ряд получен из некоторых наблюдений за системой. Т.е., если X – искомый временной ряд, то существует некоторая функция α, называемая *функцией наблюдения*, и некоторая точка $x_0 \in M$, такая, что

$$X_i = \alpha(\phi^i(x_0)).$$

В 1981 году Такенс доказал следующую теорему:

Теорема 3 (теорема Такенса [31])**.** *Если M – это компактное риманово многообразие размерности m, $\phi \in \mathrm{Diff}^2(M)$ и $\alpha \in C^2(M, \mathbb{R})$, тогда $M \to \mathbb{R}^{2m+1}$:*

$$x \mapsto (\alpha(x), \alpha \circ \phi(x), ..., \alpha \circ \phi^{2m}(x))$$

является вложением, называемым вложением Такенса.

В последствии было доказано еще несколько теорем похожего характера, с теоретическим обзором можно ознакомиться в [32].

Таким образом, благодаря теореме Такенса, существует способ реконструкции пространства состояний исходной динамической системы по набору наблюдений за ней. Такая реконструкция, конечно, не идентична исходному фазовому пространству, но тем не менее топологически эквивалентна ему. Более того, теорема Такенса дает рецепт по реконструкции: если дан временной ряд X, тогда нужно стоить $2m+1$-мерные наборы

$$R_m(t) = [X(t), X(t-1), ..., X(t-2m)],$$

и тогда совокупность этих наборов реконструирует исходное фазовое пространство. Число $2m+1$ называется *параметром размерности вложения Такенса*, а само реконструированное пространство, т.е. набор векторов $R_m(t)$, будем обозначать X_m.

Однако в случае, когда дан лишь временной ряд X, нет никакой информации о размерности подлежащего многообразия, а значит и нет оценки на размерность

наборов, которыми можно реконструировать пространство параметров. В таких ситуациях возникает задача поиска подходящего параметра размерности вложения.

4.2 Существующие методы поиска параметра размерности вложения

Для решения такой задачи Kennel et al. разработали алгоритм `False Nearest Neighbors (FNN)` [16], основанный на следующей идее.

Пусть для данного временного ряда X минимальным значением размерности вложения является m_0. Это значит, что реконструированное фазовое пространство топологически эквивалентно исходному, и значит, что окрестности точек при таком сопоставлении переходят в окрестности точек. Предположим теперь, что при помощи вложения Такенса временной ряд X оказался вложен в $\mathbb{R}^m$, где $m < m_0$. Это означает, что настоящее фазовое пространство оказалось спроецировано на $\mathbb{R}^m$, а значит топология была нарушена, и существуют такие точки, которые при проекции оказались в малой окрестности других точек, в окрестности которых они не находятся в исходном пространстве. Такие точки называют *ложными соседями*.

Итак, если существует точка p_i в m-мерном реконструированном фазовом пространстве, и ее ближайшим соседом является p_j, т.е. $||p_i - p_j|| < \varepsilon$ для некоторого $\varepsilon > 0$, тогда можно посчитать нормализованное расстояние между $(m+1)$-й координатой вложения точек p_i и p_j по следующей формуле:

$$R_i = \frac{|x_{i+m} - x_{j+m}|}{||p_i - p_j||}$$

Если R_i оказывается больше, чем некоторое пороговое значение R_{tr}, тогда p_j является *ложным ближайшим соседом* для точки p_i. Параметр порога R_{tr} подбирается экспериментально, обычно полагают, что $R_{tr} = 10$. Тогда для каждого значения размерности можно подсчитать отношение числа точек с ложным ближайших соседом ко всему числу точек. Когда это отношение достигает 0, т.е. когда ни у одной точки нет ложного ближайшего соседа, то нужная размерность найдена (рис 4.1).

4.3 Использование устойчивых гомологий для определения размерности вложения

Метод `FNN` опирается на топологию реконструирумоего фазового пространства: если нужная размерность найдена, то, вложив временной ряд на одну размерность выше, топология не изменится, отношение числа ложных точек ко всем точкам будет 0.

Но раз для нахождения нужной размерности достаточно смотреть на топологию получаемого пространства, можно воспользоваться устойчивыми гомологиями. Действительно, пусть для данного временного ряда X минимальным значением размерности вложения является m_0 и он в результате вложения оказался вложен в $\mathbb{R}^m$, где $m < m_0$. Тогда, раз m_0 – истинное значение размерности, то существует отображение $f : X_{m_0} \to X_m$, проецирующее исходное фазовое пространство (говоря точнее, реконструкцию фазового пространства, которая топологически эквивалентна ему) на пространство меньшей размерности. В гомологиях индуцированное отображение f_* будет эпиморфизмом, а значит при таком отображении какие-то циклы могут исчезнуть.

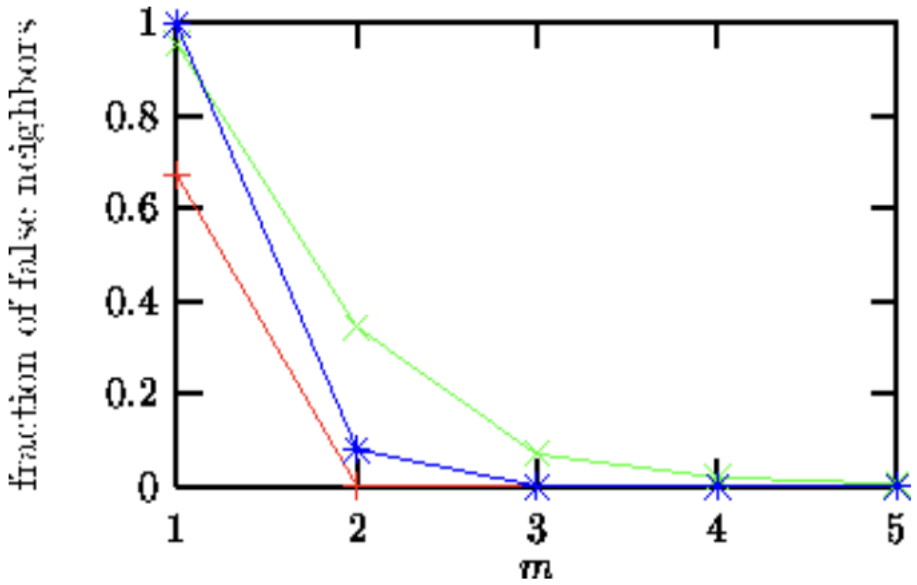


Рис. 4.1: Отношение числа точек с ложным ближайшим соседом к числу всех точек для разных временных рядов, полученных из следующих динамических систем: динамической системы Лоренца (голубая), динамической системы Генона (красная) и динамической системы Генона с 10% шумом (зеленая)

Таким образом, можно использовать понятие *топологической сложности*, о котором экстенсивно пойдет речь в главе 5. А именно, имея облако точек D, назовем его топологической сложностью $TC(D)$ число точек всех размерностей (вплоть до заранее определенного числа d, до которого вычисляются устойчивые гомологии) на диаграмме устойчивости. Из соображений выше следует, что когда найдена настоящая размерность m_0, то

$$TC(X_0) \leq TC(X_1) \leq ... \leq TC(X_{m_0}) = TC(X_{m_0+1}) = ... \, .$$

4.4 Эксперименты

Применим полученный алгоритм к 3 датасетам: `ROSSLER`, `LORENZ`, `HENON` 4.2. Это временные ряды, полученные из одноименных хаотических динамических систем. Для подсчета устойчивых гомологий воспользуемся пакетом `Ripser` на `Python` [23], а для процедуры вложения Такенса, а также нахождения параметра размерности методом `FNN` воспользуемся пакетом `Teaspoon` [33], который предназначен для топологической обработки сигналов.

В рамках эксперимента будем вычислять устойчивые гомологии вплоть до размерности 5. Аналогично, будем вкладывать временной ряд вплоть до размерности 5. Результаты вычислений устойчивых гомологий представлены на рис. 4.3.

Результаты применения метода ложных соседей показаны на рис. 4.4.

Из рассмотренных результатов видно, что алгоритм поиска параметра размерности с помощью устойчивых гомологий дает сравнимые результаты с методом ложных соседей: для датасетов `ROSSLER` и `LORENZ` метод ложных соседей оценивает подходящую размерность как 2, и на диаграмм устойчивости также видно, что после размерности 2 (второе изображение) новые гомологии не появляются: не появляется новых старших, так и не появляются новые очень устойчивые, далекие от диагонали, точки на диаграмме. В частности, топологическая сложность перестает сильно изменяться

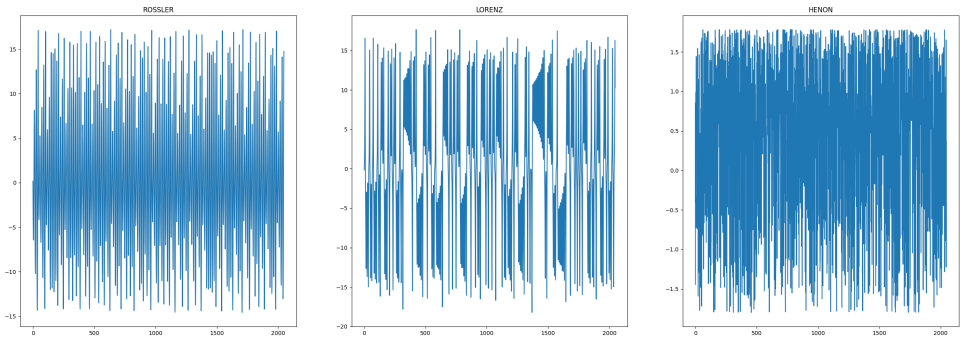


Рис. 4.2: Временные ряды, взятые для эксперимента

после размерности 2. В конце концов, из теории динамических систем известно, что корректное значение размерности для данных динамических систем равно 2.

С другой стороны, на датасете `HENON` метод на основе устойчивых гомологий, как и метод ложного соседа, не предсказывает истинное значение размерности (в случае этого датасета, корректное значение размерности равно 3). Метод ложного соседа оценивает подходящую размерность как 4, а метод на основе устойчивых гомологий не сошелся к какой-то размерности: топологическая сложность растет на каждой размерности, в старших (≥ 3) размерностях возникают нетривиальные старшие гомологии. Вероятно, что при дальнейшей вычислении устойчивых гомологий метод сошелся бы к какому-то значению.

4.5 Заключение

Таким образом, на основе устойчивых гомологий был разработан алгоритм поиска значения параметра размерности вложения Такенса временного ряда. Данный алгоритм показывает сравнимые результаты с методом ложного соседа и по существу является обобщением идеи, стоящей за этим методом.

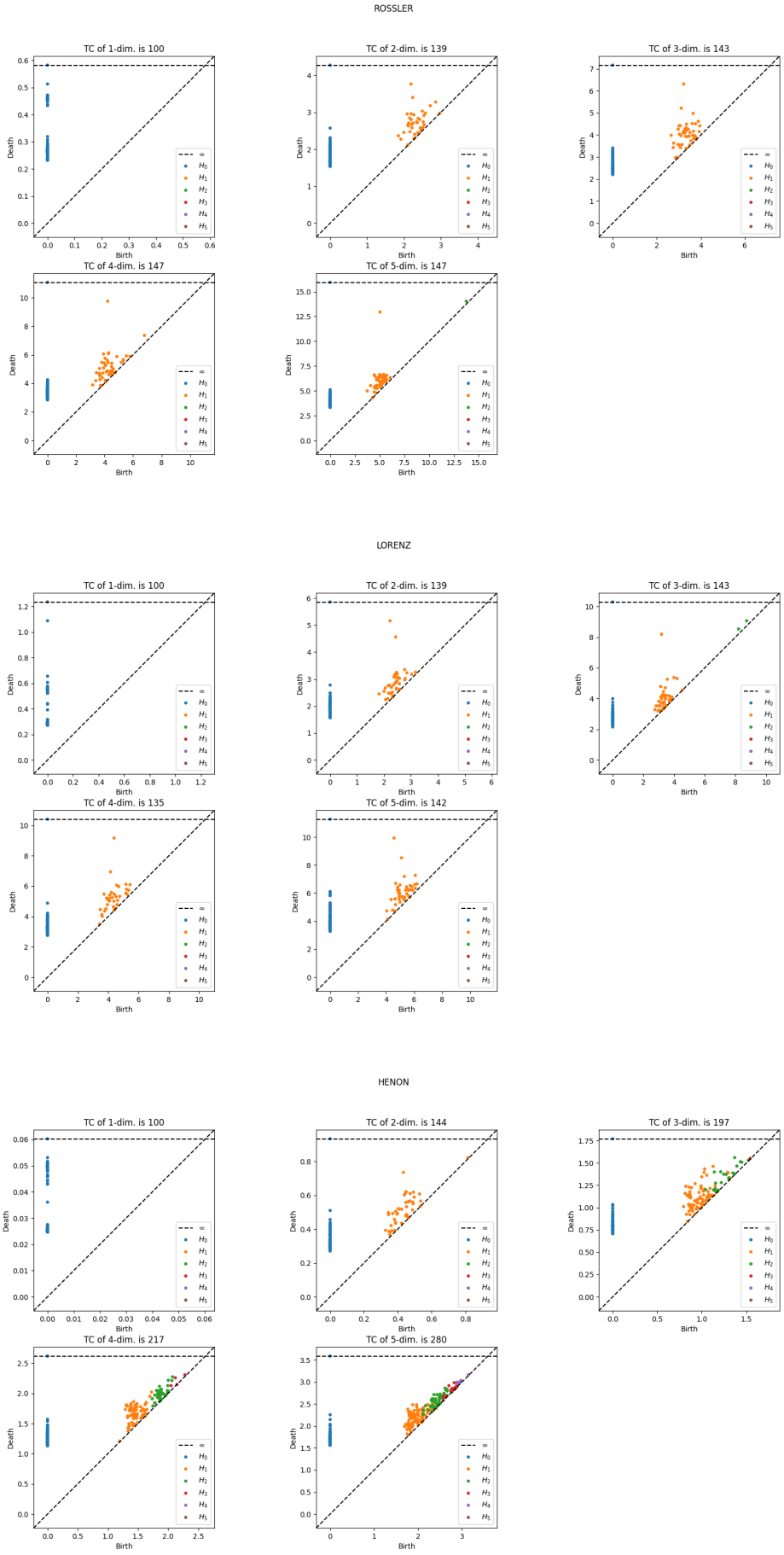


Рис. 4.3: Результаты вычислений устойчивых гомологий

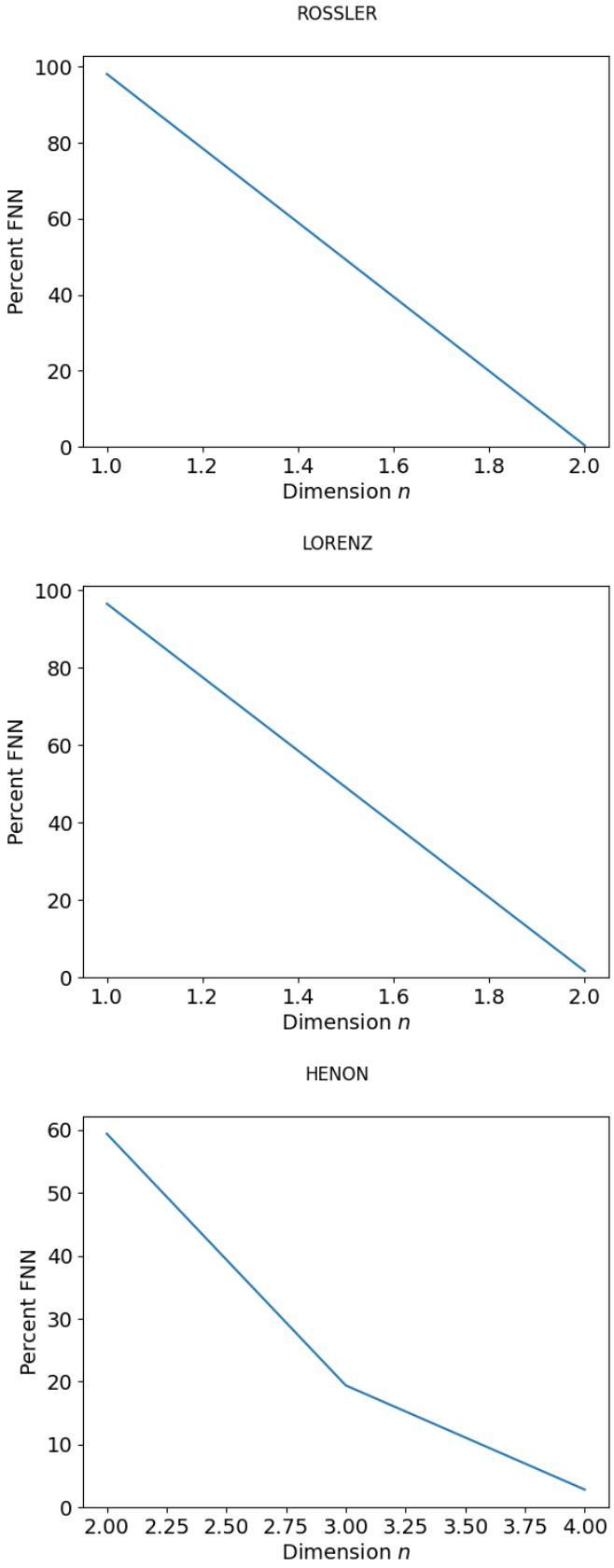


Рис. 4.4: Результаты работы метода ложных соседей

Глава 5

Топологический анализ изменений внутренних представлений данных внутри нейронных сетей

5.1 Введение

Следующая задача посвящена анализу внутреннего представления данных внутри искуственной нейронной сети. Несмотря на значительные достижения нейронных сетей в различных областях, полное объяснение механизма процесса обучения остается открытым и наиболее важным вопросом. И анализ внутреннего представления данных в нейронных сетях, анализ динамики изменения геометрии и топологии данных в скрытых слоях нейронных сетей может помочь пролить свет на механизм обучения.

Принимая гипотезу о многообразии [34], будем считать, что исходные данные лежат в окрестности некоторого многообразия, чья размерность сильно меньше размерности объемлющего пространства. Если набор данных разделен на k классов, будем считать, что исходное многообразие M, которое моделирует эти данные, представляется в виде (дизьюнктного) объединения многообразий, соответствующих каждому классу:

$$M = M_1 \cup ... \cup M_k.$$

Данная работа вдохновлена одной из основополагающих работ в данном направлении[35]. В ней авторы изучают, как топология набора данных $M = M_a \cup M_b$, представляющих два класса a и b изменяется при прохождении через слои нейронной сети, хорошо обученной для решения задачи бинарной классификации. Авторы рассматривают стандартные глубокие нейросети, т.е. такие функции $\mathrm{DNN} : \mathbb{R}^d \to [0,1]$, представленные в виде композиции

$$\mathrm{DNN} = s \circ f_l \circ f_{l-1} \circ ... f_2 \circ f_1,$$

где каждый *слой* нейросети $f_j : \mathbb{R}^{n_j} \to \mathbb{R}^{n_{j+1}}$ является композицией аффинного отображения $x \mapsto Ax+b$ и некоторой нелинейной функцией, называемой *функцией активации* $\sigma : \mathbb{R} \to \mathbb{R}$, применяемой покомпонентно. Под функцией s понимается обычный линейный классификатор. Авторы считают нейросеть хорошо обученной, если она достигает 100% точности на обучающей выборке и имеет около-нулевую ошибку на валидационной выборке.

В своей работе авторы под изменением топологии набора данных подразумевают изменение групп гомологий. А именно, назовем *топологической сложностью*

многообразия просто сумму его чисел Бетти,

$$\mathrm{TC}(M) \coloneqq \sum_{i \geq 0} \beta_i.$$

Для вычисления топологической сложности облака точек, можно воспользоваться инструментом устойчивых гомологий. А именно, *топологической сложностью* TC_t *в момент времени* t облака точек X назовем сумму чисел Бетти симплициального комплекса – объкта фильтрации в момент времени t. *Устойчивой топологической сложностью* облака точек назовем последовательность $TC(X) \coloneqq \{TC_t(X)\}$.

Авторы провели обширную работу по изучению изменения топологии в том числе и реальных данных, и сделали следующий выбор: нейронная сеть оперирует изменением топологии данных, трансформируя «топологически сложный»(используя понятие топологической сложности, введенное выше) датасет в «топологически тривиальный» (т.е. $\mathrm{TC}(X) = 0$), как только он проходит через слои нейронной сети. Такое изменение в топологии происходит значительно быстрее в нейронных сетях, использующих ReLU в качестве функции активации. Более того, неглубокие нейронные сети оперируют топологией в последних слоях, в то время как глубокие нейронные сети равномерно распределяют изменения топологии по всем слоям.

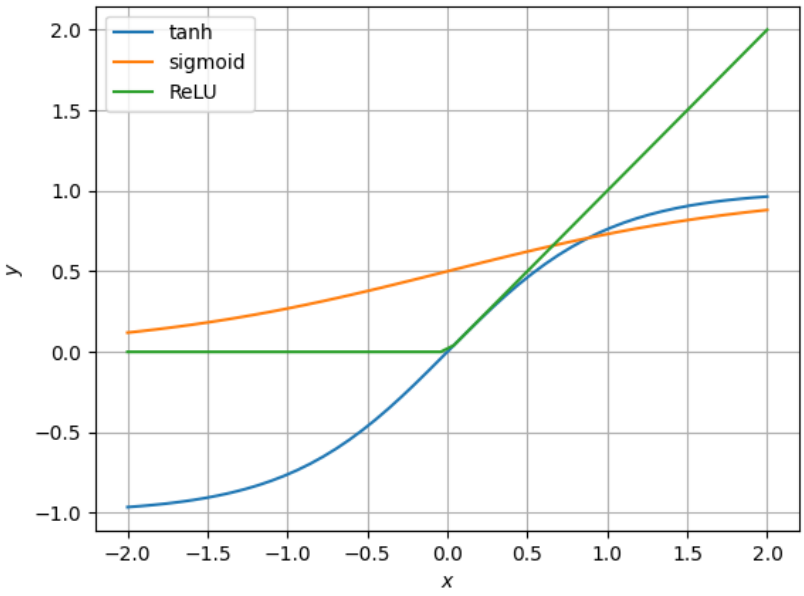


Рис. 5.1: Графики самых распространенных функций активации

Таким образом, авторы статьи дают новую перспективу на превосходство ReLU над другими стандартными функциями активации, в частности над гиперболическим тангенсом (рис. 5.1). В отличие от последнего, ReLU не является гомеоморфизмом, а значит и меняет топологию. Действительно, по определению,

$$\mathrm{ReLU}(x) = \begin{cases} 0, \text{если } x < 0, \\ x, \text{иначе.} \end{cases}$$

А значит ReLU не инъективна, поэтому не является гомеоморфизмом, «склеивает» между собой точки, которые меньше нуля. В то время как гиперболический тангенс tanh является гомеоморфизмом. Этот факт дает меньше возможностей нейросети как-либо изменить топологию данных. Тем не менее, даже с гиперболическим

тангенсом в качестве функции активации топология может меняться. Более того, не меняется она только в одном случае

Лемма 1 ([36]). *Пусть L – слой нейронной сети с N входами и N выходами, а также с гиперболическим тангенсом в качестве функции активации, т.е.*

$$L(x) = \tanh(Ax + b).$$

Тогда L – гомеоморфизм, тогда и только тогда, когда матрица A не является вырожденной, т.е. $\det(A) \neq 0$.

Таким образом, ReLU предоставляет дополнительную степень свободы в отношении изменения топологии, благодаря чему упрощение данных происходит быстрее. А раз нейросеть, решая задачу (бинарной) классификации, максимально упрощает топологию данных, то чем быстрее происходит такое упрощение, тем быстрее нейросеть обучается.

Исходя из этого, можно предложить использовать в качестве функций активации другие отображения, не сохраняющие топологию. Более того, раз упрощение топологии является главной задачей, которую решает нейросеть в процессе обучения, то можно попытаться построить новые функции активации, которые могут позволить нейросети более эффективно упрощать топологию.

5.2 Другие работы, изучающие топологию внутренних представлений данных

Помимо работы [35], существуют также и другие работы, изучающие динамику и изменения геометрических и топологических свойств внутренних представлений данных. Так, в работе [37] показано, что процесс обучения глубоких нейронных сетей связан с процессом распутывания скрытых многообразий, на которых лежат данные, при прохождении через скрытые слои нейросети. Авторы работы [38] изучают то, как изменяется геометрия скрытого многообразия (его размерность и т.п.) в процессе обучения, и приходят к выводу, что при прохождении через хорошо обученную нейросеть скрытые многообразия, на которых лежат данные разных классов, на выходе из нейросети становятся линейно разделимыми. В работе [39] изучаются преобразования данных в глубокой нейронной сети и предпринимаются попытки формулирования теории глубокого обучеия в терминах Римановой геометрии. В [40] анализируется способность глубокой нейронности сети выучивать эффективное маломерное представление.В [41] авторы используют устойчивые ландшафты [42] – технику векторизации диаграмм устойчивости – для анализа динамики топологической сложности при прохождении через скрытые слои нейронной сети, и показывают, что топологические характеристики не всегда упрощаются в процессе обучения. В работе [43] изучаются внутренние представления в различных современных нейронных сетях, предназначенные для компьютерного зрения или обработки естественного языка с точки зрения топологии и наблюдается эффект, что процесс обучения данных нейронных сетей также непосредственно связан с изменением топологических характеристик, а также показано, что качество и обобщающая способность моделей может быть связана с топологией и геометрией данных.

Данная же работа вдохновлена работой [35] и является ее продолжением. Далее будет предложена другая функция активации, которая, как и ReLU не является гомеоморфизмом, и будет изучена динамика топологической сложности внутренних представлений данных по мере прохождения данных через скрытые слои нейросети.

5.3 Негомеоморфные функции активации

ReLU является неинъективной функцией, она сжимает данные, тем самым изменяя топологию и «убивая» соответствующие нетривиальные циклы в гомологиях. Но этого же эффекта можно добиться, «разрезая» исходное многообразие по нетривиальным циклам. В целом, можно разделять исходное многообразие, разрезая его, и эта операция, конечно, сильно меняет топологию. В качестве такой функции, разрезающей многообразие можно рассмотреть несюръективную функцию с точкой разрыва с конечным скачком, например функцию

$$\text{split-sign}(x) = x + \text{sign}(x) \cdot c.$$

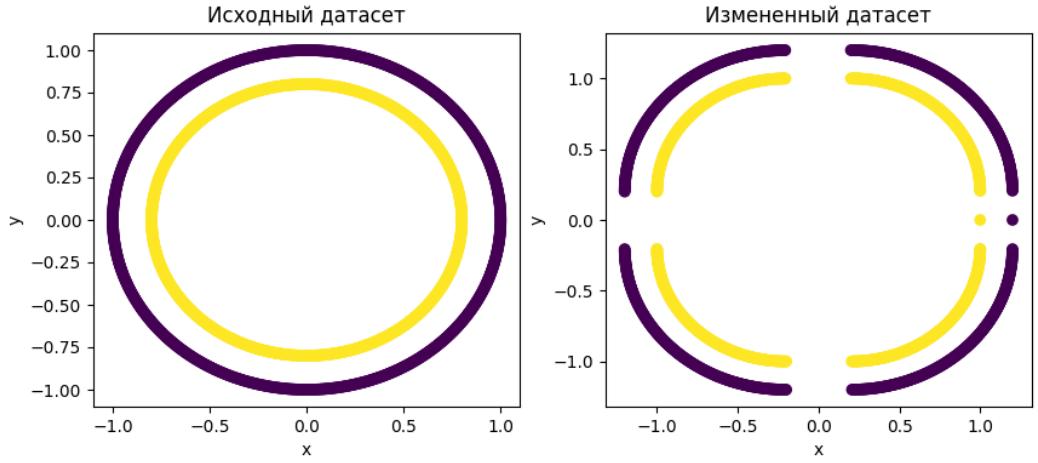


Рис. 5.2: Пример изменения топологии с помощью разрывной функции split-sign

Такая функция «разделяет» $\mathbb{R}$ в точке 0, тем самым изменяя топологию вещественной прямой. Если использовать такую функцию как функцию активации, исходное многообразие данных будет разделено по каждому из измерений (рис. 5.2).

Можно модифицировать данную функцию, добавив гиперпараметр, отвечающий за сдвиг точки раздела,

$$\text{split-sign}(x) = x + \text{sign}((x - a)) \cdot c.$$

К сожалению, такая функция не является дифференцируемой, и потому она не может выступать в качестве функции активации в нейронной сети. Поэтому воспользуемся ее гладкими приближениями, например,

$$\text{split-tanh} = x + \tanh\big(\lambda(x - a)\big) \cdot c.$$

При достаточно большом значении параметра λ получаемая функция очень точно приближает искомую функцию split-sign(x) (рис. 5.3), и при этом является гладкой, а значит может быть использована в качестве функции активации.

Полученная функция split-tanh на самом деле является небольшой модификацией хорошо знакомого механизма в глубоком обучении, а именно *skip (или residual) connection*,

$$y = F(x) + x,$$

где в качестве F может выступать, например, сам слой нейронной сети (т.е. композиция $x \mapsto \sigma(Ax + b)$), так и сама функция активации. В нашем случае в качестве F выступает функция

$$x \mapsto \tanh(\lambda(x - a)) \cdot c,$$

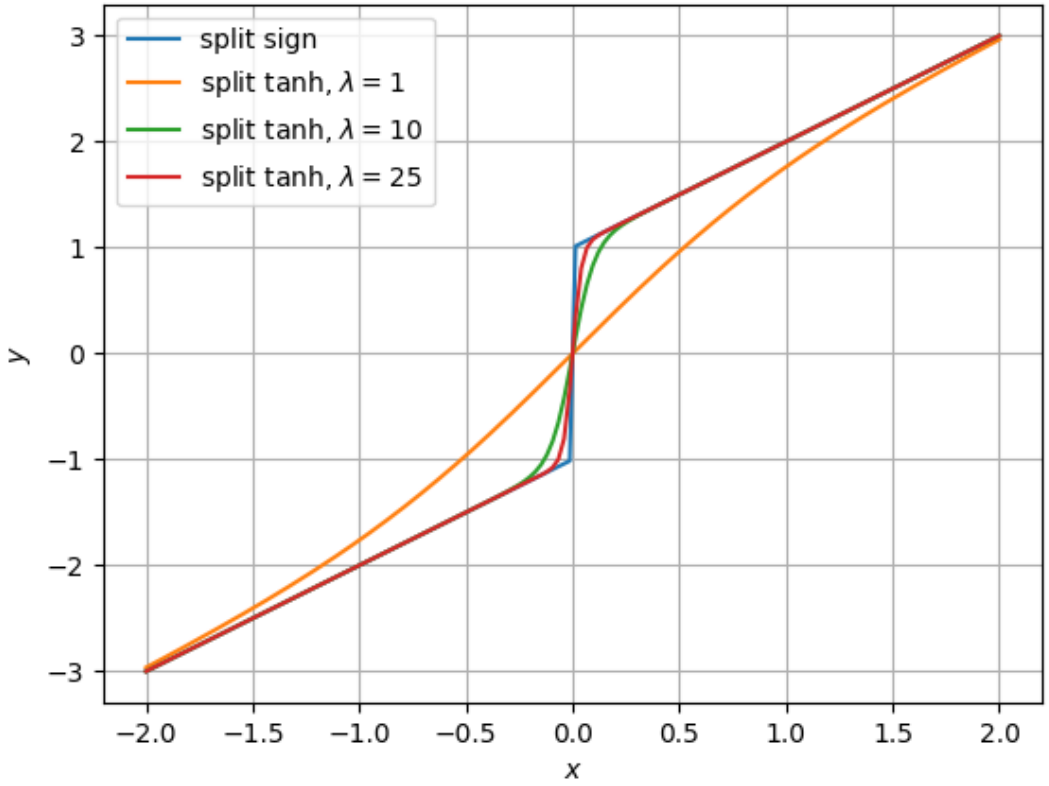


Рис. 5.3: Приближения функции split-sign функцией split-tanh с различными значениями λ

что можно рассматривать как дополнительный слой нейросети, где в качестве афинного отображения $x \mapsto Ax + b$ рассматривается поокординатное отображение $x \mapsto \lambda x - \lambda a$ (или, что эквивалентно, в качестве A выступает диагональная матрица подходящего размера с параметром λ на диагонали).

Механизм skip connection появился еще в 1997 году в работах Юргена Шмидхубера [44]. Таким образом такая формулировка данного механизма позволяет взглянуть по-другому на него по-другому, предоставляет топологическую формулировку и объяснение причины его успеха.

Также можно рассмотреть следующую функцию :

$$\text{split-trig}(x) = \begin{cases} bx + b\cos a - \sin a, & \text{если } x \leq -\cos a \\ x\,\text{tg}(a), & \text{если } -\cos a \leq x \leq \cos a \\ x + \sin a - \cos a, & \text{если } x \geq \cos a \end{cases}$$

Данная функция примечательна тем, что при подходящих параметрах она будет моделировать либо ReLU, либо split-sign, либо split-tanh, а именно

- При $a = 0, b = 0, \text{split-trig}(x) = \text{ReLU}(x - 1)$
- При $a = \frac{\pi}{2}, b = 1, \text{split-trig}(x) = \text{split-sign}(x)$
- При $a \in [\frac{\pi}{4}, \frac{\pi}{2}), b = 1, \text{split-trig}(x) = \text{split-tanh}(x)$, причем параметр λ функции split-tanh однозначно определен по параметру a (см. рис. 5.4).

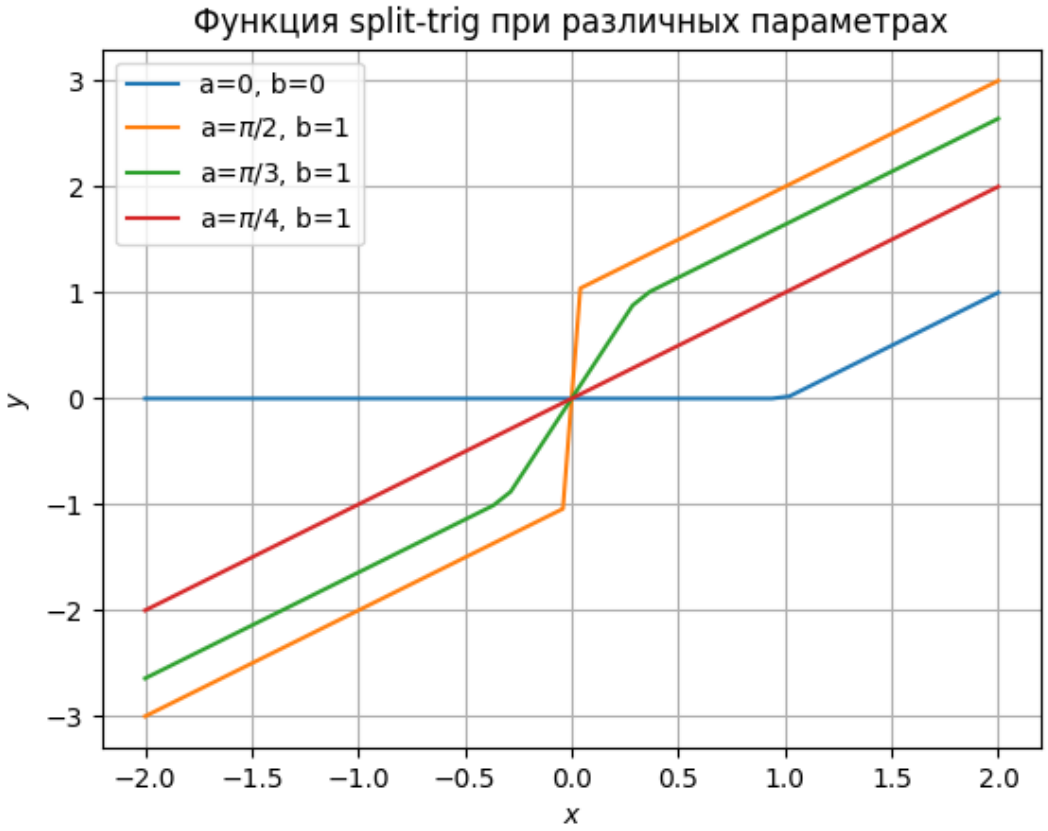


Рис. 5.4: Функция split-trig при различных параметрах

То есть таким образом, вместо функции ReLU, которая является примером неинъективного непрерывного отображения, можно рассматривать несюръективное непрерывное отображение, например split-sign или split-tanh. Такая замена предопределяет характер топологических преобразований с исходными данными или с многообразием, на котором эти данные находятся. В первом случае часть многообразия склеивается в одну точку, благодаря чему и происходят топологические изменения. При использовании же таких несюръективных отображений исходное многообразие «разделяется», что также приводит к изменению топологии. Функция же split-trig позволяет комбинировать эти два подхода. Имеющиеся параметры, в свою очередь, можно настраивать в процессе обучения, так как данные функции дифференцируемы относительно неизвестных параметров.

5.4 Методология и эксперименты

Далее изучим, насколько хорошо можно использовать такие функции в качестве функций активации в нейронных сетях, сравним их с ReLU. Также проанализируем, как изменяется топология данных при прохождении через слои нейронной сети с той или иной функцией активации. В силу вычислительных аспектов, сравненивать с ReLU будем исключительно функцию split-tanh.

5.4.1 Используемые датасеты

Сгенерируем 3 набора данных: `circles, tori, disks` 5.5.

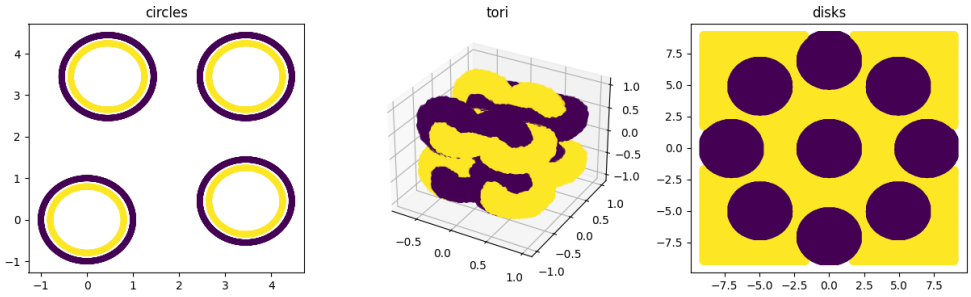


Рис. 5.5: Многообразия, подлежащие под датасетами

Датасет `circles` семплирован из одномерного многообразия $M = M_a \cup M_b$, каждый из классов содержит 4 компоненты связности, где каждая компонента гомеоморфна S^1. Датасет `tori` семплирован из двумерного многообразия $M = M_a \cup M_b$, где каждая из компонент гомеоморфна тору $S^1 \times S^1$. Датасет `disks` семплирован из двумерного многообразия $M = M_a \cup M_b$, где один из классов представляет собой связную, но не односвязную область, а другой состоит из нескольких двумерных дисков.

5.4.2 Архитектура и обучение нейронных сетей

Для построения и обучения нейросетей воспользуемся пакетом `PyTorch` [45].

Будем анализировать нейросети:

- с различной *глубиной* – различным числом скрытых слоев,
- с различной *шириной* – числом нейронов в скрытых слоях.

Конкретные архитектуры показаны в табл. 5.1 и 5.2. Во 2-ом столбце таблиц указана функция активации, которая используется во всех слоях, кроме последних двух: последний слой представляет собой обычный линейный классификатор, а на предпоследнем слое используется ReLU. Это используется для того, чтобы, следуя выводам из [35], «склеить» куски многообразия, которые были разделены с помощью split-tanh. В последнем столбце указано число экспериментов – запусков процесса обучения и подсчета устойчивых гомологий внутренних представлений – с конкретной архитектурой.

Данные разобьем на обучающую и валидационную выборки в соотношении 5:1. В качестве функции потерь используется бинарная кросс-энтропия. В качестве метода оптимизации используется метод `Adam` [46] с темпом обучения (learning rate) равным $3e-3$ и числом эпох равным 5000.

Таблица 5.1: Архитектуры нейронных сетей

Датасеты	Активация	# нейронов в каждом слое	# экспериментов
circles	ReLU	2-3-3-3-2	30
circles	split-tanh	2-3-3-3-2	30
circles	ReLU	2-5-5-5-2	30
circles	split-tanh	2-5-5-5-2	30
circles	ReLU	2-7-7-7-2	30
circles	split-tanh	2-7-7-7-2	30
circles	ReLU	2-9-9-9-2	30
circles	split-tanh	2-9-9-9-2	30
circles	ReLU	2-3-3-3-3-3-2	30
circles	split-tanh	2-3-3-3-3-3-2	30
circles	ReLU	2-5-5-5-5-5-2	30
circles	split-tanh	2-5-5-5-5-5-2	30
circles	ReLU	2-7-7-7-7-7-2	30
circles	split-tanh	2-7-7-7-7-7-2	30
circles	ReLU	2-9-9-9-9-9-2	30
circles	split-tanh	2-9-9-9-9-9-2	30
circles	ReLU	2-3-3-3-3-3-3-3-2	30
circles	split-tanh	2-3-3-3-3-3-3-3-2	30
circles	ReLU	2-5-5-5-5-5-5-5-2	30
circles	split-tanh	2-5-5-5-5-5-5-5-2	30
circles	ReLU	2-7-7-7-7-7-7-7-2	30
circles	split-tanh	2-7-7-7-7-7-7-7-2	30
circles	ReLU	2-9-9-9-9-9-9-9-2	30
circles	split-tanh	2-9-9-9-9-9-9-9-2	30
tori	ReLU	3-3-3-3-2	30
tori	split-tanh	3-3-3-3-2	30
tori	ReLU	3-5-5-5-2	30
tori	split-tanh	3-5-5-5-2	30
tori	ReLU	3-7-7-7-2	30
tori	split-tanh	3-7-7-7-2	30
tori	ReLU	3-9-9-9-2	30
tori	split-tanh	3-9-9-9-2	30
tori	ReLU	3-3-3-3-3-3-2	30
tori	split-tanh	3-3-3-3-3-3-2	30
tori	ReLU	3-5-5-5-5-5-2	30
tori	split-tanh	3-5-5-5-5-5-2	30
tori	ReLU	3-7-7-7-7-7-2	30
tori	split-tanh	3-7-7-7-7-7-2	30
tori	ReLU	3-9-9-9-9-9-2	30
tori	split-tanh	3-9-9-9-9-9-2	30
tori	ReLU	3-3-3-3-3-3-3-3-2	30
tori	split-tanh	3-3-3-3-3-3-3-3-2	30
tori	ReLU	3-5-5-5-5-5-5-5-2	30
tori	split-tanh	3-5-5-5-5-5-5-5-2	30
tori	ReLU	3-7-7-7-7-7-7-7-2	30
tori	split-tanh	3-7-7-7-7-7-7-7-2	30
tori	ReLU	3-9-9-9-9-9-9-9-2	30
tori	split-tanh	3-9-9-9-9-9-9-9-2	30

Таблица 5.2: Архитектуры нейронных сетей: продолжение

Датасеты	Активация	# нейронов в каждом слое	# экспериментов
disks	ReLU	2-3-3-3-2	30
disks	split-tanh	2-3-3-3-2	30
disks	ReLU	2-5-5-5-2	30
disks	split-tanh	2-5-5-5-2	30
disks	ReLU	2-7-7-7-2	30
disks	split-tanh	2-7-7-7-2	30
disks	ReLU	2-9-9-9-2	30
disks	split-tanh	2-9-9-9-2	30
disks	ReLU	2-3-3-3-3-3-2	30
disks	split-tanh	2-3-3-3-3-3-2	30
disks	ReLU	2-5-5-5-5-5-2	30
disks	split-tanh	2-5-5-5-5-5-2	30
disks	ReLU	2-7-7-7-7-7-2	30
disks	split-tanh	2-7-7-7-7-7-2	30
disks	ReLU	2-9-9-9-9-9-2	30
disks	split-tanh	2-9-9-9-9-9-2	30
disks	ReLU	2-3-3-3-3-3-3-3-2	30
disks	split-tanh	2-3-3-3-3-3-3-3-2	30
disks	ReLU	2-5-5-5-5-5-5-5-2	30
disks	split-tanh	2-5-5-5-5-5-5-5-2	30
disks	ReLU	2-7-7-7-7-7-7-7-2	30
disks	split-tanh	2-7-7-7-7-7-7-7-2	30
disks	ReLU	2-9-9-9-9-9-9-9-2	30
disks	split-tanh	2-9-9-9-9-9-9-9-2	30

5.4.3 Анализ изменения топологии данных при прохождении через слои нейросети

Для анализа изменения топологии данных воспользуемся инструментом устойчивых гомологий. А именно будем смотреть на то, как меняются устойчивые 1-гомологии данных при прохождении через слои нейросети. Так, если $\mu : \mathbb{R}^d \to [0,1]$ – нейросеть с l скрытыми слоями, то нас будут интересовать $PH_1(\mu_j(X))$, где μ_j – это j-ый слой нейросети μ. Имея $PH_1(\mu_j(X))$, посчитаем (устойчивую) топологическую сложность

$$TC_1(\mu_j(X)) = \#\{PH_1(\mu_j(X))\}.$$

Фокус на устойчивых 1-гомологиях не случаен: в сравнении участвуют две функции активации, ReLU и split-tanh, причем эти две функции по-разному изменяют топологию: ReLU сжимает подмногообразия, тогда как split-tanh разделяет исходное многообразие на два новых многообразия. Разделение исходного многообразия, несомненно, отразится на топологии уже тем, что в 0-гомологиях, которые отвечают за компоненты связности, появятся новые генераторы. В свою же очередь, у ReLU такого эффекта нет. Таким образом, включая в эксперимент также подсчет устойчивых 0-гомологий, эффект от потенциального упрощения топологии внутренних представлений данных при использовании split-tanh в качестве функции активации был бы не так заметен при подсчете топологической сложности, и тем самым та-

кой топологический дескриптор не корректно отражал бы наличие или отсутствие упрощения топологии данных.

Подсчет устойчивых 1-гомологий в ходе экспериментов осуществим с помощью пакета `Ripser` на языке `Python` [23], который, в свою очередь, использует пакет для быстрых вычислений устойчивых гомологий `Ripser`, написанный на `C++` [22].

5.5 Результаты и заключение

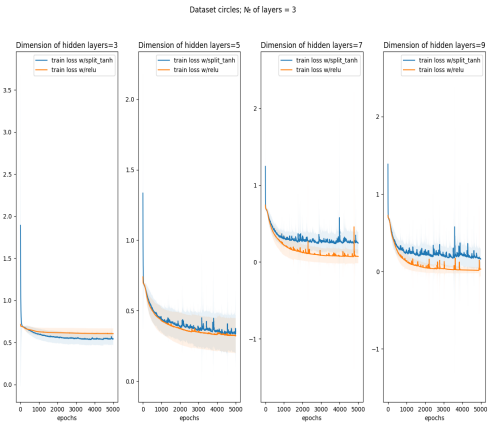
Таким образом, механизм эксперимента следующий:

1. Генерируется датасет X, сэмплированный с некоторого многообразия $M = M_a \cup M_b \subseteq \mathbb{R}^d$;

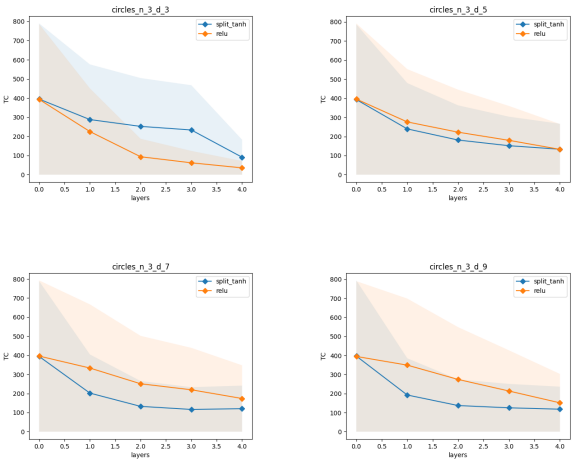
2. Обучается нейросеть с l скрытыми слоями $\mu : \mathbb{R}^d \to [0,1]$ для решения задачи бинарной классификации на сгенерированном датасете;

3. Используя обученную нейросеть, считаются (устойчивые) 1-гомологии выхода с j-го слоя нейросети $\mu_j(X)$, вычисляется топологическая сложность.

Результаты такого эксперимента изображены на рис. 5.6 - 5.14.

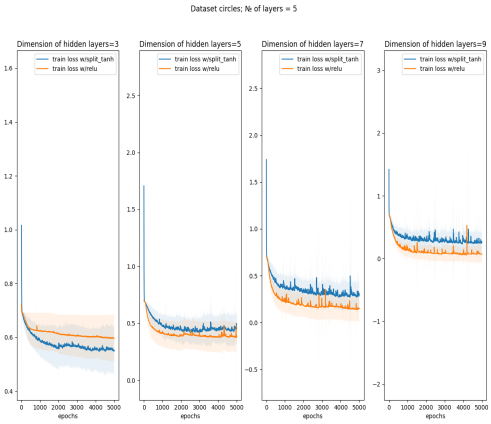


(a) Лосс при обучении

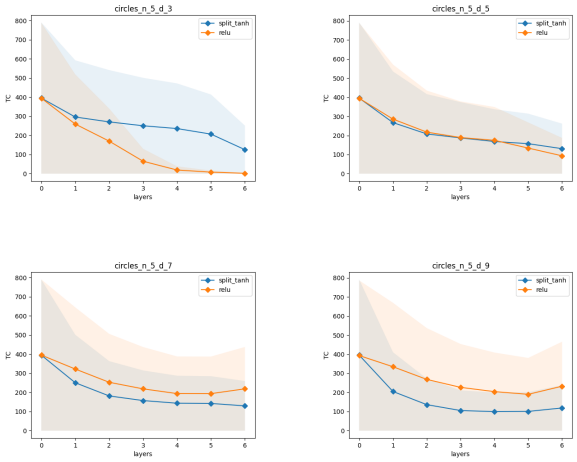


(b) Изменения топологической сложности внутренних представлений

Рис. 5.6: Результаты эксперимента для датасета `circles`; число скрытых слоев нейросети $l = 3$

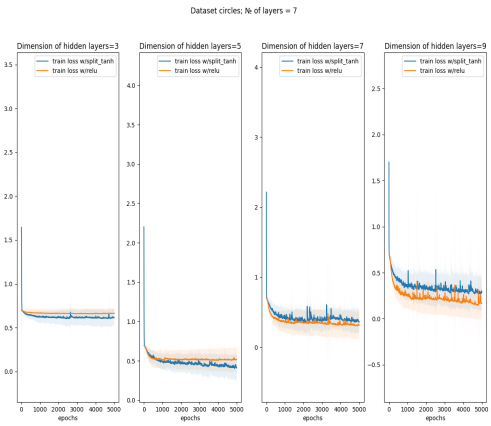


(a) Лосс при обучении

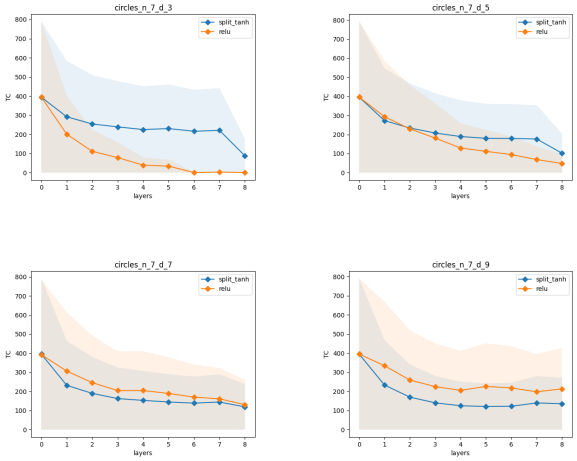


(b) Изменения топологической сложности внутренних представлений

Рис. 5.7: Результаты эксперимента для датасета `circles`; число скрытых слоев нейросети $l = 5$

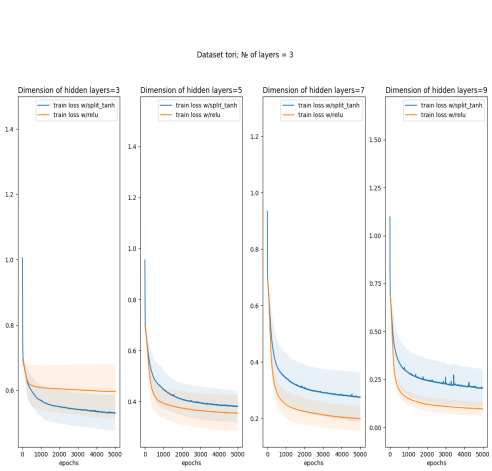


(a) Лосс при обучении

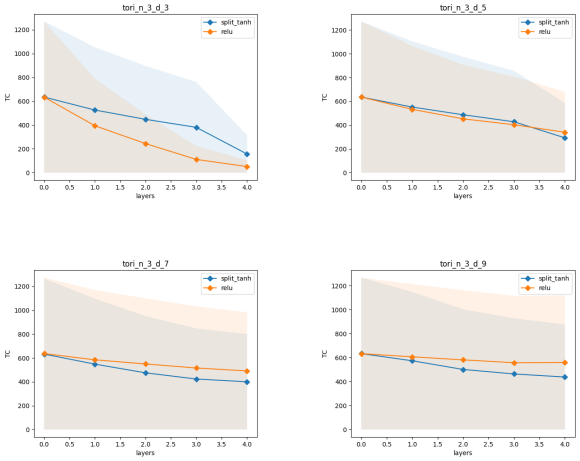


(b) Изменения топологической сложности внутренних представлений

Рис. 5.8: Результаты эксперимента для датасета `circles`; число скрытых слоев нейросети $l = 7$

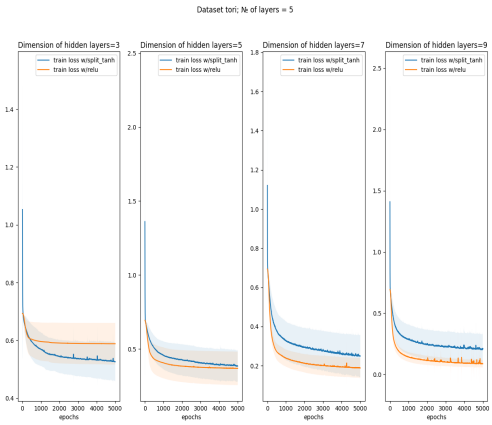


(a) Лосс при обучении

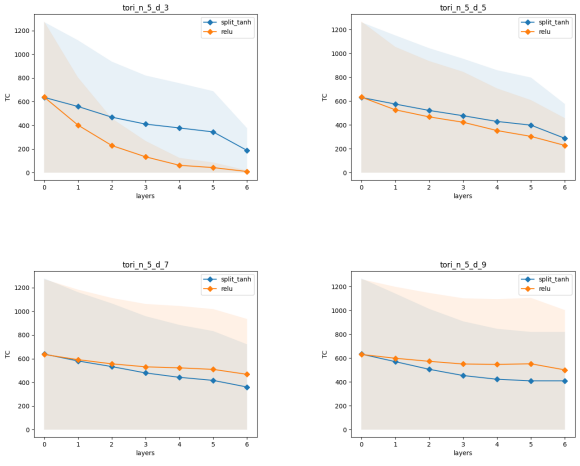


(b) Изменения топологической сложности внутренних представлений

Рис. 5.9: Результаты эксперимента для датасета `tori`; число скрытых слоев нейросети $l = 3$

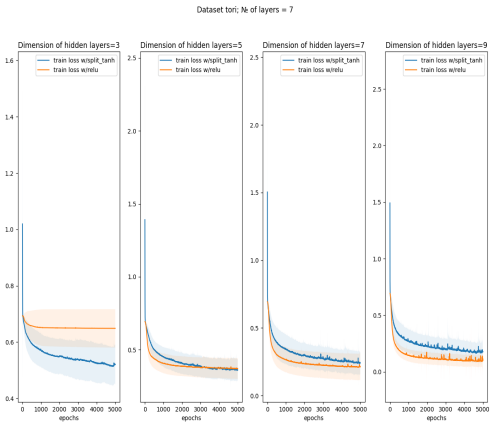


(a) Лосс при обучении

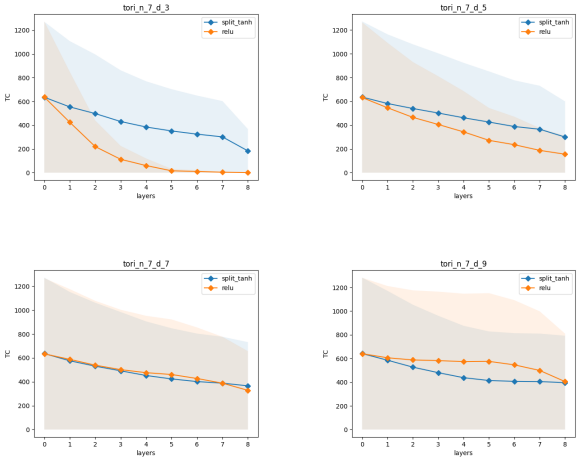


(b) Изменения топологической сложности внутренних представлений

Рис. 5.10: Результаты эксперимента для датасета `tori`; число скрытых слоев нейросети $l = 5$

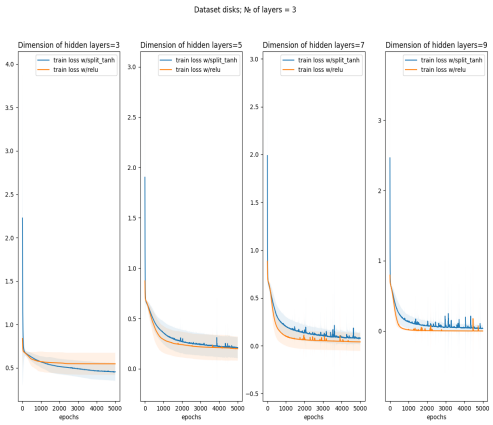


(a) Лосс при обучении

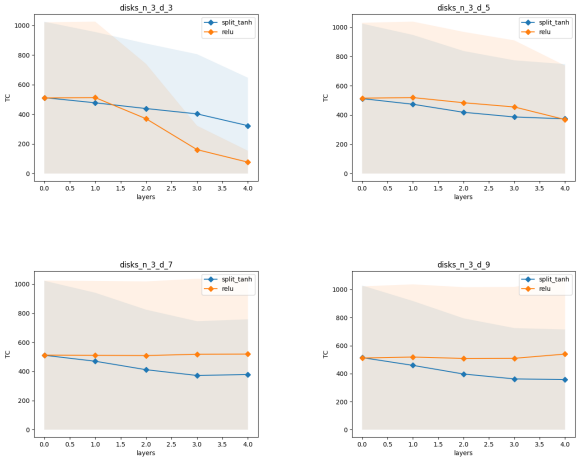


(b) Изменения топологической сложности внутренних представлений

Рис. 5.11: Результаты эксперимента для датасета `tori`; число скрытых слоев нейросети $l = 7$

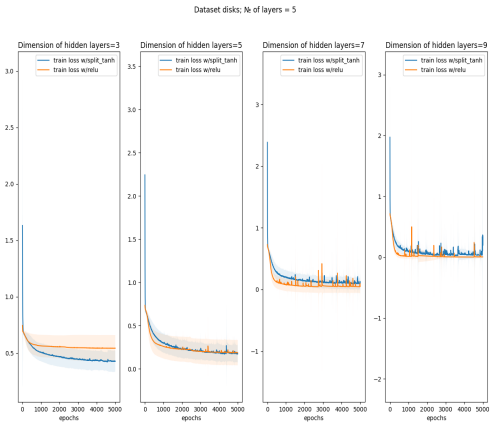


(a) Лосс при обучении

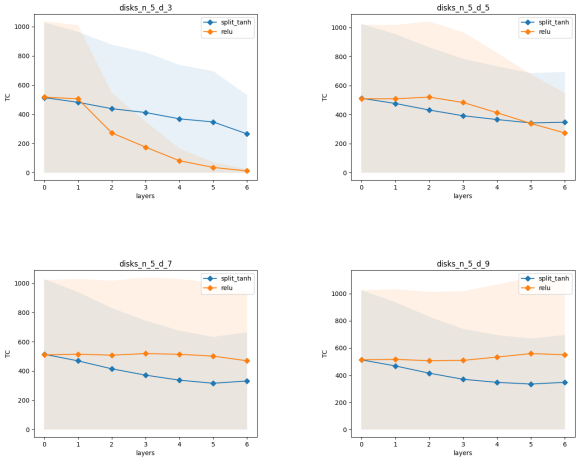


(b) Изменения топологической сложности внутренних представлений

Рис. 5.12: Результаты эксперимента для датасета `disks`; число скрытых слоев нейросети $l = 3$

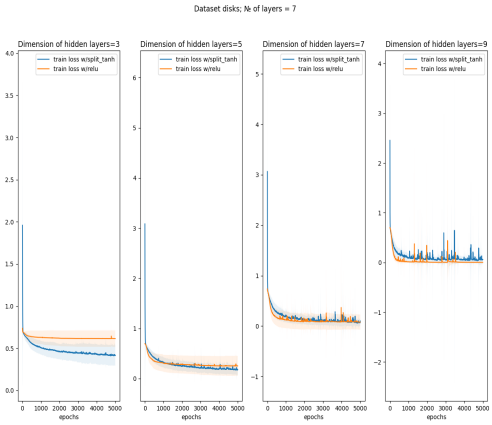


(a) Лосс при обучении

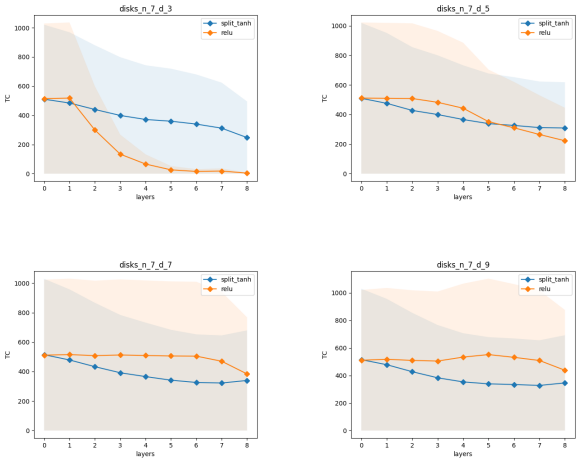


(b) Изменения топологической сложности внутренних представлений

Рис. 5.13: Результаты эксперимента для датасета `disks`; число скрытых слоев нейросети $l = 5$



(a) Лосс при обучении



(b) Изменения топологической сложности внутренних представлений

Рис. 5.14: Результаты эксперимента для датасета `disks`; число скрытых слоев нейросети $l = 7$

Исходя из результатов, можно сделать следующие выводы:

- Действительно, чем лучше нейросеть обучена, тем лучше она упрощает топологию данных;
- Наблюдается следующий эффект: низкому качеству модели на обучении сопутствует резкое упрощение топологии на первых слоях;
- При этом плавное снижение топологии сопутствует более высокому качеству модели на обучении;
- Модели с split-tanh в качестве функции активации в скрытых слоях в среднем лучше упрощают топологию данных на протяжении всех слоев за исключением тех случаев, где ReLU сильно упрощает топологию в первых слоях, т.е.

за исключением тех случаев, где качество модели с ReLU в качестве функции активации низкое;

- Также модели с split-tanh в качестве функции активации обучаются лучше, чем модели с ReLU в качестве функции активации в ситуации, когда число нейронов в слое небольшое;
- При этом при увеличении размерности внутренного представления модели с ReLU в качестве функции активации на обучении показывают более лучшее качество.

Таким образом, использование негомеоморфных функций активации, действительно, сильнее позволяет нейросетям упрощать топологию данных, что позволяет таким нейросетям быстрее обучаться. При этом, такие функции активации могут не только сжимать исходные данные или многообразия, на котором эти данные лежат, как это делает ReLU, но и разделять их, тем самым упрощая топологию. Именно таким образом работает функция активации split-tanh, которая является в некоторой степени модификацией известного в глубоком обучении механизма skip connection. В результате экспериментов было показано, что такая функция может успешно применяться в качестве функции активации в нейросетях, в частности для задач (бинарной) классификации. Использование такой функции позволяет модели быстрее упрощать топологию данных, распределяя нагрузку по упрощению топологии равномерно между всеми слоями нейросети, что приводит к более стабильному обучению. Помимо этого, нейронные сети с такой функцией активации показывают гораздо более лучшее качество обучения при небольшом числе нейронов в скрытых слоях модели.

Глава 6

Заключение

В данной работе были рассмотрены различные применения устойчивых гомологий в задачах анализа данных. Были продемонстрированы возможности, которые открываются при использовании этого инструмента в частности, а также топологического анализа данных в целом.

В главе 2 был изложен теоретический материал по алгебре и топологии, а также определены устойчивые гомологии (как функторы $\mathbf{T} \to \mathbb{Z}-\mathbf{Mod}$ из категории $\mathbf{T}$, соответствующей частично упорядоченному множеству $T \subseteq \mathbb{R}$ в категорию модулей над $\mathbb{Z}$, или, что то же самое, в категорию абелевых групп. Были изложены необходимые теоремы и конструкции, которые и используются на практике. Также были обозрены методы вычислений устойчивых гомологий на практике.

В главе 3 была решена задача симплификации кривой. Был получен непараметрический алгоритм на основе устойчивых гомологий для решения данной задачи, а также были проведены сравнения качества работы данного алгоритма с алгоритмом Рамера-Дугласа-Пекера, являющегося наилучшим решением данной задачи. Было показано, что алгоритм на основе устойчивых гомологий способен решать данную задачу не сильно хуже алгоритма Рамера-Дугласа-Пекера, при этом являясь непараметрическим методом, что сильно упрощает его использование.

В главе 4 была решена задача поиска параметра размерности вложения Такенса временного ряда.

В главе 5 был проведен анализ изменений внутренних представлений данных внутри нейронных сетей. Было изучено, как различные функции активации могут влиять на топологию внутренних представлений. Было показано, что хорошообученная нейронная сеть, решающая задачу (бинарной) классификации упрощает топологию данных. Была предложена модификация известного механизма skip connection в качестве функции активации, изучено его влияние на изменение топологии внутренних представлений. Такой анализ демонстрирует новый, основанный на геометрии и топологии, подход к изучению и объяснению процесса обучения нейронных сетей.

В рамках будущей работы, можно продолжить исследование топологии внутренних представлений данных внутри нейронных сетей. Можно использовать другие топологические дескрипторы, помимо топологической сложности, и отслеживать их изменение. Вполне вероятно, что существуют такие числовые характеристики геометрии или топологии внутренних представлений данных, которые могут не меняться от слоя к слою. Также интересно изучить влияние на топологию данных других широко распространненных механизмов обучения нейронных сетей (например, dropout). Огромный интерес вызывает изучение топологии внутренних представлений более сложных нейронных сетей, используемых в настоящее время в областях

компьютерного зрения или обработки естественных языков: сверточных нейронных сетей, Трансформеров.

Список использованных источников

1. Edelsbrunner H., Harer J. Computational Topology An Introduction. — Providence : AMS, 2009. — 241 p.
2. Zomorodian A., Carlsson G. Computing Persistent Homology // Discrete Comput. Geom. — 2005. — Vol. 33, no. 2. — P. 249–274.
3. Barannikov S. The Framed Morse complex and its invariants // Advances in Soviet Mathematics. — 1994. — Апр. — Т. 21. — С. 93—116. — (Singularities and Bifurcations). — URL: https://hal.science/hal-01745109.
4. Xia K., Wei G.-W. Persistent homology analysis of protein structure, flexibility, and folding. // Int J Numer Method Biomed Eng. — 2014. — Авг. — Т. 30, № 8. — С. 814—844.
5. Duman A. N., Pirim H. Gene Coexpression Network Comparison via Persistent Homology. // Int J Genomics. — Department of Mathematics и др., 2018. — Т. 2018. — С. 7329576.
6. Topological analysis of interaction patterns in cancer-specific gene regulatory network: persistent homology approach / H. Masoomy [и др.] // Scientific Reports. — 2021. — Т. 11, № 1. — С. 16414. — URL: https://doi.org/10.1038/s41598-021-94847-5.
7. Persistent homology-based functional connectivity and its association with cognitive ability: Life-span study / H. Ryu [и др.] // Human Brain Mapping. — 2023. — Т. 44, № 9. — С. 3669—3683. — URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.26304.
8. Persistent Brain Network Homology From the Perspective of Dendrogram / H. Lee [и др.] // IEEE transactions on medical imaging. — 2012. — Сент. — Т. 31.
9. Di Fabio B., Landi C. A Mayer–Vietoris Formula for Persistent Homology with an Application to Shape Recognition in the Presence of Occlusions // Foundations of Computational Mathematics. — 2011. — Т. 11, № 5. — С. 499—527. — URL: https://doi.org/10.1007/s10208-011-9100-x.
10. Silva V., Ghrist R., Muhammad A. Blind Swarms for Coverage in 2-D //. — 06.2005. — С. 335—342.
11. Horak D., Maletić S., Rajković M. Persistent homology of complex networks // Journal of Statistical Mechanics: Theory and Experiment. — 2009. — Март. — Т. 2009, № 03. — P03034. — URL: https://doi.org/10.1088%2F1742-5468%2F2009%2F03%2Fp03034.
12. On the Local Behavior of Spaces of Natural Images / G. Carlsson [и др.] // International Journal of Computer Vision. — 2008. — Янв. — Т. 76. — С. 1—12.
13. Topology-Based Kernels With Application to Inference Problems in Alzheimer's Disease / D. Pachauri [и др.] // IEEE transactions on medical imaging. — 2011. — Апр. — Т. 30. — С. 1760—70.

14. Frosini P., Landi C. Persistent Betti Numbers for a Noise Tolerant Shape-Based Approach to Image Retrieval //. T. 34. — 01.2011. — C. 294—301.

15. Fractal and Computational Geometry for Generalizing Cartographic Objects / O. Musin [и др.] // Modeling and Analysis of Information Systems. — 2012. — Т. 19, № 6. — С. 152—160.

16. Kennel M. B., Brown R., Abarbanel H. D. I. Determining embedding dimension for phase-space reconstruction using a geometrical construction // Phys. Rev. A. — 1992. — Март. — Т. 45, вып. 6. — С. 3403—3411. — URL: https://link.aps.org/doi/10.1103/PhysRevA.45.3403.

17. Weibel C. An Introduction to Homological Algebra. — Cambridge University Press, 1994. — (Cambridge Studies in Advanced Mathematics). — URL: https://books.google.ru/books?id=flm-dBXfZ%5C_gC.

18. Riehl E. Category Theory in Context. — Dover Publications, 2017. — (Aurora: Dover Modern Math Originals). — URL: https://books.google.ru/books?id=6B9MDgAAQBAJ.

19. Schenck H. Algebraic Foundations for Applied Topology and Data Analysis. — Springer International Publishing, 2022. — (Mathematics of Data). — URL: https://books.google.ru/books?id=4IQkzwEACAAJ.

20. Crawley-Boevey W. Decomposition of pointwise finite-dimensional persistence modules. — 2014.

21. Oudot S. Persistence Theory: From Quiver Representations to Data Analysis. — 01.2015.

22. Bauer U. Ripser: efficient computation of Vietoris-Rips persistence barcodes // J. Appl. Comput. Topol. — 2021. — Т. 5, № 3. — С. 391—423. — URL: https://doi.org/10.1007/s41468-021-00071-5.

23. Tralie C., Saul N., Bar-On R. Ripser.py: A Lean Persistent Homology Library for Python // The Journal of Open Source Software. — 2018. — Сент. — Т. 3, № 29. — С. 925. — URL: https://doi.org/10.21105/joss.00925.

24. The GUDHI Project. GUDHI User and Reference Manual. — 3.4.1. — GUDHI Editorial Board, 2021.

25. Čufar M. Ripserer.jl: flexible and efficient persistent homology computation in Julia // Journal of Open Source Software. — 2020. — Т. 5, № 54. — С. 2614. — URL: https://doi.org/10.21105/joss.02614.

26. Айзенберг А. Методичка по симплициальным комплексам и гомологиям.

27. Kozlov Y., Weinkauf T. Persistence1d. — 2014. — URL: https://www.csc.kth.se/~weinkauf/notes/persistence1d.html.

28. TopoLines: Topological Smoothing for Line Charts / P. Rosen [и др.]. — 2020.

29. Milnor J. Morse Theory. (AM-51), Volume 51. — Princeton University Press, 1969. — URL: http://www.jstor.org/stable/j.ctv3f8rb6 (дата обр. 08.06.2023).

30. Yahoo Finance. — URL: https://finance.yahoo.com.

31. Takens F. Detecting strange attractors in turbulence // Dynamical Systems and Turbulence, Warwick 1980 / под ред. D. Rand, L.-S. Young. — Berlin, Heidelberg : Springer Berlin Heidelberg, 1981. — С. 366—381.

32. Sauer T., Yorke J. A., Casdagli M. Embedology // Journal of Statistical Physics. — 1991. — Т. 65, № 3. — С. 579—616. — URL: https://doi.org/10.1007/BF01053745.

33. Munch L., Khasawneh F. Teaspoon: Topological Signal Processing in Python. — URL: https://teaspoontda.github.io/teaspoon/installation.html.

34. Manifold Hypothesis in Data Analysis: Double Geometrically-Probabilistic Approach to Manifold Dimension Estimation / A. Ivanov [и др.]. — 2021.

35. Naitzat G., Zhitnikov A., Lim L.-H. Topology of Deep Neural Networks // J. Mach. Learn. Res. — 2020. — Янв. — Т. 21, № 1.

36. Olah C. Neural Networks, Manifolds, and Topology. — 2014. — URL: https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/.

37. Brahma P., Wu D., She Y. Why Deep Learning Works: A Manifold Disentanglement Perspective // IEEE Transactions on Neural Networks and Learning Systems. — 2015. — Дек. — Т. 27. — С. 1—12.

38. Separability and geometry of object manifolds in deep neural networks / U. Cohen [и др.] // Nature Communications. — 2020. — Т. 11, № 1. — С. 746. — URL: https://doi.org/10.1038/s41467-020-14578-5.

39. Hauser M., Ray A. Principles of Riemannian Geometry in Neural Networks // Advances in Neural Information Processing Systems. Т. 30 / под ред. I. Guyon [и др.]. — Curran Associates, Inc., 2017. — URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/0ebcc77dc72360d0eb8e9504c78d38bd-Paper.pdf.

40. Basri R., Jacobs D. Efficient Representation of Low-Dimensional Manifolds using Deep Networks. — 2016.

41. Wheeler M., Bouza J., Bubenik P. Activation Landscapes as a Topological Summary of Neural Network Performance // 2021 IEEE International Conference on Big Data (Big Data). — IEEE, 12.2021. — URL: https://doi.org/10.1109%5C%2Fbigdata52589.2021.9671368.

42. Bubenik P., Dłotko P. A persistence landscapes toolbox for topological statistics // Journal of Symbolic Computation. — 2017. — Т. 78. — С. 91—114. — URL: https://www.sciencedirect.com/science/article/pii/S0747717116300104 ; Algorithms and Software for Computational Topology.

43. Magai G. Deep neural networks architectures from the perspective of manifold learning. — 2023.

44. Hochreiter S., Schmidhuber J. Long Short-term Memory // Neural computation. — 1997. — Дек. — Т. 9. — С. 1735—80.

45. PyTorch: An Imperative Style, High-Performance Deep Learning Library / A. Paszke [и др.] // Advances in Neural Information Processing Systems 32. — Curran Associates, Inc., 2019. — С. 8024—8035. — URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

46. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization. — 2017.

47. TISEAN. Nonlinear Time Series Analysis package. False Nearest Neighbors page. — URL: https://www.pks.mpg.de/tisean/TISEAN_2.1/docs/chaospaper/node9.html.

48. Mileyko Y., Mukherjee S., Harer J. Probability measures on the space of persistence diagrams // Inverse Problems. — 2011. — Vol. 27, no. 12.