

Министерство науки и высшего образования РФ

Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского

МАТЕМАТИКА И ИТ — ВМЕСТЕ В ЦИФРОВОЕ БУДУЩЕЕ

Сборник трудов III Молодежной школы

Нижний Новгород, 22–26 апреля 2024 г.

Нижний Новгород
ННГУ им. Н.И. Лобачевского

2024

УДК 004.942(06)+519.876.5(06)
ББК 22.18я43
М34

М34 Математика и ИТ – вместе в цифровое будущее. Сборник трудов III Молодежной школы (Н. Новгород, 22–26 апреля 2024 г.) / Под ред. К.А. Баркалова, М.А. Богомоловой. – Нижний Новгород: Нижегородский госуниверситет им. Н.И. Лобачевского, 2024. –70 с.

Отв. за выпуск Т.Г. Смирнова

ISBN 978-5-91326-904-1

Сборник трудов участников конкурса научно-исследовательских проектов по математике и информационным технологиям III Молодежной школы «Математика и ИТ – вместе в цифровое будущее», состоявшейся 22–26 апреля 2024 г. на базе Нижегородского государственного университета им. Н.И. Лобачевского, содержит тезисы докладов и короткие статьи, тематика которых посвящена информационным и сквозным технологиям цифровой экономики, математическому моделированию сложных процессов и явлений, численным методам их исследования.

ISBN 978-5-91326-904-1

УДК 004.942(06)+519.876.5(06)
ББК 22.18я43

© Коллектив авторов, 2024
© ННГУ им. Н.И. Лобачевского, 2024

Предисловие

22–26 апреля 2024 г. в Институте информационных технологий, математики и механики ННГУ им. Н.И. Лобачевского состоялась III Молодежная школа «Математика и ИТ — вместе в цифровое будущее». Участники школы встретились с учеными Университета Лобачевского и ведущими практиками из российских и международных ИТ-компаний в онлайн-формате. В общении со студентами приняли участие компаний Т1, Гарда Технологии, ЛАНИТ Омни, Неофлекс и Норбит.

Программа молодежной школы включала следующие лекции:

- Константинов Александр (ННГУ им. Н.И. Лобачевского, д.ф.-м.н., профессор): «Инженерное обеспечение цифрового проектирования конструкций»;
- Тихомиров Сергей (ННГУ им. Н.И. Лобачевского, лаборант-исследователь): «Большие лингвистические модели»;
- Хорошилова Людмила (Т1, главный инженер-тестировщик): «Математика в тестировании программного обеспечения»;
- Шильникова Анастасия (Гарда Технологии, руководитель группы тестирования): «QA-инженеры в процессе разработки: как выглядит/строится процесс тестирования на практике»;
- Савватеев Алексей (Московский физико-технический институт, д.ф.-м.н., профессор): «От ловли шпионов к геометрии Лобачевского»;
- Калистратов Константин (ЛАНИТ Омни, технический владелец продукта): «Цена ошибки в IT-проекте»;
- Пригодич Илья (Неофлекс): «Системная аналитика. Какого это быть системным аналитиком, и какие инструменты используются»;
- Соболев Никита (Независимый open-source разработчик): «4 уровня знания Python»;
- Тимаков Дмитрий (НОРБИТ, руководитель направления AI): «Простыми словами, что такое коммерческий проект с применением ИИ, и за что уже реально готов платить бизнес».

В рамках работы Молодежной школы дополнительно проводился конкурс научно-исследовательских проектов по математике и информационным технологиям, тематика которых охватывала следующие направления:

- суперкомпьютерные технологии;
- искусственный интеллект;
- машинное обучение и анализ данных;
- компьютерное зрение;
- робототехника;
- обработка естественных языков;
- виртуальная реальность;
- дополненная реальность;
- интернет вещей;
- большие данные;
- математическое моделирование;
- экспериментальная механика;
- компьютерная механика;
- численные методы;

- алгебра, геометрия, теория чисел;
- теория вероятностей и приложения;
- дискретная математика;
- дифференциальные уравнения.

Всего на молодежную школу поступило более 100 заявок от студентов, обучающихся по направлениям информационных технологий и математики, более чем из 10 вузов Приволжского федерального округа. Студенты из разных городов России представили результаты своих исследований и разработок. В соответствии с тематикой мероприятия были представлены результаты как чисто математических работ, так и работ в сфере IT.

Лучшие работы были отмечены призовыми местами. Победителем конкурса стал студент НИУ «Высшая школа экономики» Рябов Кирилл. Второе место занял студент ННГУ им. Н.И. Лобачевского Уткин Герман. Третье место разделили студенты ННГУ им. Н.И. Лобачевского Пинежанин Евгений и Зорин Олег.

Подвижная конструкция, составленная из замечательных математических кривых

В.А. Бородина, О.А. Монахова

Пензенский государственный университет

В данной работе содержится описание построения подвижной конструкции, составленной из замечательных математических кривых, в интерактивной системе «GeoGebra». По авторской задумке, движение кривых позволит наблюдать смену времени суток: восход солнца, распускание цветков на лужайке, полет бабочки, заход солнца, появление созвездия в небе. Демонстрация такой конструкции студентам математической специальности (или самостоятельное выполнение ими подобной работы) формирует мотивацию к изучению замечательных кривых и возможностей различных программ динамической геометрии.

Ключевые слова: подвижная конструкция, замечательные кривые, «GeoGebra», система координат, параметр.

1. Введение

Мир замечательных математических кривых весьма многообразен. Сами их названия (конхоида Никомеда, спираль Архимеда, лемниската Бернулли, улитка Паскаля, кривые Хабенихта, овалы Кассини, спирали Галилея, Ферма и др.) свидетельствуют об интересе к ним математиков всех времен и народов. Имея различное происхождение (в основном, из задач физики), они всегда восхищали поклонников точных наук возможностью построения на их основе всевозможных оригинальных конструкций.

Сегодня в свободном доступе хранится большое количество исследовательских работ, информационных и творческих проектов, связанных с изучением математических кривых и созданием на их основе орнаментов, натюрмортов, пейзажей. Однако не существует ни одной подвижной конструкции, составленной из замечательных кривых. А между тем демонстрация такой конструкции студентам математической специальности (или самостоятельное выполнение ими подобной работы) формирует перспективную мотивацию к изучению и замечательных кривых, и возможностей программ динамической геометрии.

2. Основная часть

Была задумана следующая подвижная конструкция: «Восходит солнце, озаряя лужайку. Взору представляется пробуждающаяся природа: распускаются цветы, взлетает первая бабочка. Весь день она парит в воздухе, а к вечеру спускается на лужайку. Цветы закрываются, и на небе появляется яркое созвездие» (см. пример на рис. 1).



Рис. 1. Прообраз подвижной конструкции

После изучения замечательных кривых были обозначены следующие элементы композиции:

- 1) $y \leq k$ – полуплоскость с границей («лужайка»);
- 2) $y > k$ – полуплоскость без границы («небо»);
- 3) $(x - a)^2 + (y - b)^2 \leq c^2$ – окружность («солнце»);
- 4) $\rho = d \sin 8\varphi$ – роза Гранди («цветок»);
- 5) $\rho = 1 + 7 \cos 2\varphi + 4 \sin^2 2\varphi + 3 \sin^4 2\varphi$ – кривая Хабенихта («крылышки бабочки»);
- 6) $\rho = 2f^2 \cos 2\varphi$ – лемниската Бернулли («тельце бабочки»);
- 7) $\begin{cases} x = g \cos^3 \varphi, \\ y = g \sin^3 \varphi, \end{cases} 0 \leq \varphi \leq 2\pi$ – астроида («звезда») [2–4].

Описанную конструкцию было решено составить в интерактивной системе «GeoGebra», поскольку по сравнению с другими общедоступными программами динамической геометрии она обладает рядом преимуществ.

Во-первых, эта программа является кроссплатформенной: она может быть установлена на различные операционные системы (Windows, Linux, MacOS), есть возможность установить программу на смартфоны и планшеты, работающие под управлением операционных систем iOS и Android OS, а при необходимости – работать онлайн на сайте. Среда «GeoGebra» распространяется бесплатно, что позволяет беспрепятственно устанавливать ее на компьютер дома и в учебных заведениях.

Во-вторых, программа «GeoGebra» предоставляет пользователям широкие возможности для построения геометрических объектов, создания анимированных компьютерных моделей, графического решения уравнений, их систем, построения графиков функций за счет команд встроенного языка (который также позволяет управлять и геометрическими построениями). Фактически программное обеспечение «GeoGebra» визуализирует связь между алгеброй, которая изучает буквенно-числовые выражения, и геометрией, которая изучает фигуры, их свойства, взаимное расположение.

Следует заметить, что создание задуманной конструкции даже в передовом программном обеспечении «GeoGebra» представляет определенные трудности. Они связаны с построением кривых, заданных в полярных координатах и параметрически (в программе «GeoGebra» можно построить кривые, заданными прямоугольными координатами), выбором допустимых диапазонов вариации параметров. Преодоление их сводится к описанию процедур для изображения каждой отдельной кривой.

Итак, для построения кривых, заданных в полярных координатах, потребовалось осуществить переход от полярной системы координат к декартовой. Для этого использовались известные отношения:

$$x = \rho \cos \varphi,$$

$$y = \rho \sin \varphi,$$

$$\rho = \sqrt{x^2 + y^2},$$

$$\varphi = \arctg \frac{y}{x} \text{ (см. геометрическую интерпретацию на рис. 2) [1].}$$

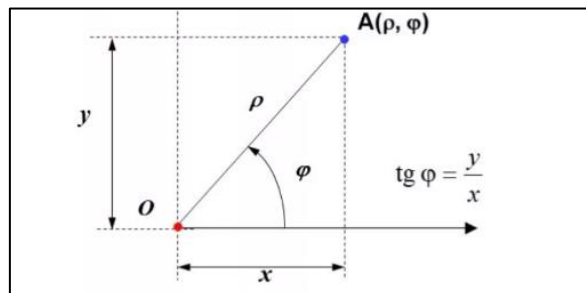


Рис. 2. Связь между декартовой и полярной системами координат

В качестве примера рассмотрим получение уравнения «цветка» при переходе от полярной системы координат к декартовой:

$$\rho = d \sin 8\varphi,$$

$$\rho = 2d \sin 4\varphi \cos 4\varphi,$$

$$\rho = 4d \sin 2\varphi \cos 2\varphi \cos 4\varphi,$$

$$\rho = 8d \sin \varphi \cos \varphi \cos 2\varphi \cos 4\varphi,$$

$$\rho = 8d \sin \varphi \cos \varphi (1 - 2\sin^2 \varphi)(1 - 2\sin^2 2\varphi),$$

$$\rho = 8d \sin \varphi \cos \varphi (1 - 2\sin^2 \varphi)(1 - 8(\sin \varphi \cos \varphi)^2),$$

$$\sqrt{x^2 + y^2} = 8d \frac{y}{\sqrt{x^2 + y^2}} \frac{x}{\sqrt{x^2 + y^2}} \left(1 - 2\left(\frac{y}{\sqrt{x^2 + y^2}}\right)^2\right) \left(1 - 8\left(\frac{y}{\sqrt{x^2 + y^2}} \frac{x}{\sqrt{x^2 + y^2}}\right)^2\right),$$

$$\sqrt{x^2 + y^2} = \frac{8dxy}{x^2 + y^2} \left(1 - \frac{2y^2}{x^2 + y^2}\right) \left(1 - \frac{8x^2y^2}{(x^2 + y^2)^2}\right).$$

Аналогично были получены следующие уравнения:

$$\sqrt{x^2 + y^2} = 8 - \frac{14y^2}{x^2 + y^2} + \frac{16x^2y^2}{(x^2 + y^2)^2} + \frac{48x^4y^4}{(x^2 + y^2)^4} - \text{кривая Хабенихта («крылышки бабочки»);}$$

$$(x^2 + y^2)^2 = 2f^2(x^2 - y^2) - \text{лемниската Бернулли («тельце бабочки»);}$$

$$x^{\frac{2}{3}} + y^{\frac{2}{3}} = g^{\frac{2}{3}} - \text{астроида («звезда»)}.$$

Создание конструкции началось с изображения «лужайки» и «неба». Для этого в программе «GeoGebra» были графически решены неравенства $y \leq 12$ и $y > 12$ (значение $k = 12$ было выбрано так, чтобы впоследствии при заданном диапазоне параметров «солнце» не опускалось ниже верхней полуплоскости, «неба»). Были использованы зеленый и голубой цвета заливки.

Затем было необходимо рассмотреть движение кривых в программе динамической геометрии, обусловленное допустимыми диапазонами вариации параметров.

Так, было графически решено неравенство:

$$(x - a)^2 + (y - b)^2 \leq c^2.$$

Была осуществлена заливка круга желтым цветом. В этом неравенстве параметр a отвечает за перемещение круга вдоль оси y , параметр b – за его перемещение вдоль оси x , параметр c – за длину радиуса окружности. В результате одновременного уменьшения значений параметров a и b и c круг движется, «опускается» по дуге, постепенно уменьшаясь («солнце скрывается, наступает вечер»).

В следующем уравнении величина параметра d ($d \in [0; 5]$) влияет на размер «цветка»: с уменьшением d цветок «закрывается», с увеличением – «распускается»:

$$\sqrt{x^2 + y^2} = \frac{8dxy}{x^2 + y^2} \left(1 - \frac{2y^2}{x^2 + y^2}\right) \left(1 - \frac{8x^2y^2}{(x^2 + y^2)^2}\right).$$

На «лужайке» было изображено три «цветка». Их уравнения выглядят следующим образом:

$$\sqrt{x^2 + y^2} = \frac{8dxy}{x^2 + y^2} \left(1 - \frac{2y^2}{x^2 + y^2}\right) \left(1 - \frac{8x^2y^2}{(x^2 + y^2)^2}\right),$$

$$\sqrt{(x - 20)^2 + (y + 15)^2} = \frac{8d(x - 20)(y + 15)}{(x - 20)^2 + (y + 15)^2} \left(1 - \frac{2(y + 15)^2}{(x - 20)^2 + (y + 15)^2}\right) \left(1 - \frac{8(x - 20)^2(y + 15)^2}{((x - 20)^2 + (y + 15)^2)^2}\right),$$

$$\sqrt{(x + 20)^2 + (y + 15)^2} = \frac{8d(x + 20)(y + 15)}{(x + 20)^2 + (y + 15)^2} \left(1 - \frac{2(y + 15)^2}{(x + 20)^2 + (y + 15)^2}\right) \left(1 - \frac{8(x + 20)^2(y + 15)^2}{((x + 20)^2 + (y + 15)^2)^2}\right).$$

К построенным кривым автоматически применилась заливка разными цветами. Ее можно по желанию изменить. Применить заливку к внутренней области не позволяют функциональные возможности программы.

Уравнения «крылышек бабочки» и «тельца бабочки» после введения параметров k и l ($k, l \in [-5; 5]$) приняли соответственно вид:

$$\sqrt{(x - k)^2 + (y - l)^2} = 8 - \frac{14(y - l)^2}{(x - k)^2 + (y - l)^2} + \frac{16(x - k)^2(y - l)^2}{((x - k)^2 + (y - l)^2)^2} + \frac{48(x - k)^4(y - l)^4}{((x - k)^2 + (y - l)^2)^4},$$

$((x - k)^2 + (y - l)^2)^2 = 9((y - l)^2 - (x - k)^2)$ (значение $2f^2 = 9$ было выбрано так, чтобы две кривые образовывали цельный объект – «бабочку»).

Смена значений параметров k и l приводит «бабочку» в движение. Ее полет символизирует смену времени суток: с восходом «солнца» она взмывает в небо, а затем начинает опускаться на «цветок». Наступает вечер, на небе появляется созвездие.

Были заданы следующие уравнения «звезд»:

$$(x - 30 + m)^{\frac{2}{3}} + (y - 30 + n)^{\frac{2}{3}} = g^{\frac{2}{3}},$$

$$(x - 25 + m)^{\frac{2}{3}} + (y - 25 + n)^{\frac{2}{3}} = g^{\frac{2}{3}},$$

$$(x - 20 + m)^{\frac{2}{3}} + (y - 20 + n)^{\frac{2}{3}} = g^{\frac{2}{3}}.$$

Были определены следующие диапазоны вариации параметров: $m \in [-5; 5]$ и $n, g \in [0; 3]$. С уменьшением m и увеличением n и g «звезды» появляются на небе и движутся таким же образом, что и «солнце» днем. Когда они полностью исчезают, начинается восход: появляется «солнце», постепенно увеличивается и поднимается все выше.

Затем начинается новый цикл (см. полученную конструкцию на рис. 3).

Изменяя скорость воспроизведения, можно увеличить количество циклов, совершаемых той или иной кривой (например, не один, а несколько полетов «бабочки» за день).

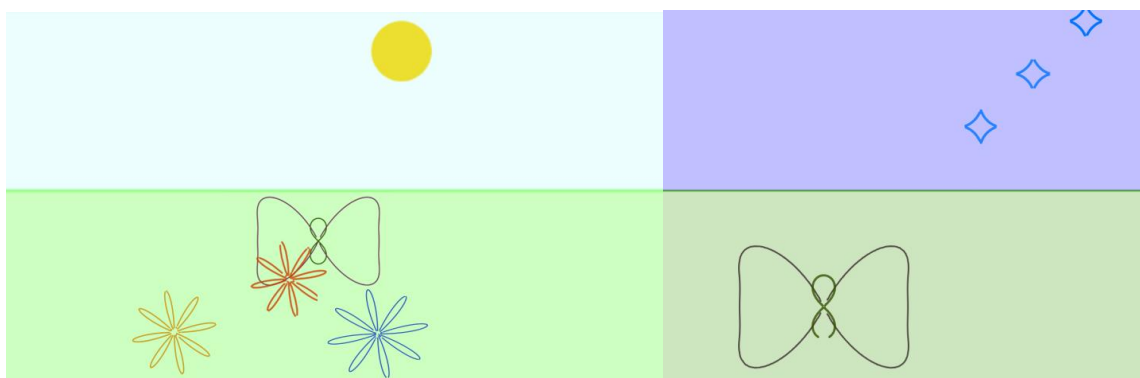


Рис. 3. Построенная конструкция

3. Заключение

Составленная подвижная конструкция полностью соответствует первоначальной задумке. По желанию можно дополнить ее новыми элементами, например, кривой в виде стрекозы. Ее уравнение ввел Хабенихт:

$$\rho = 2(1 + \cos 3\varphi) - 4\sin^2 \varphi.$$

В программе «GeoGebra» можно менять элементы дизайна (цвет заливки, толщину и вид линии и др.), получая таким образом любой задуманный узор, орнамент, изображение.

Работа по созданию такой конструкции и ее анализу способствует развитию воображения, углублению математических знаний, формированию умений применять информационные технологии в образовательном процессе.

Литература

1. Гельфанд И.М., Глаголева Е.Г., Кириллов А.А. Метод координат. М.: МЦНМО, 2009. 182 с.
2. Лавров Г.О., Лаврова Н.Г. Замечательные кривые: розы Гвидо // Международный школьный научный вестник. 2018. № 3. С. 68–77.
3. Маркушевич А.И. Замечательные кривые. М.: Наука, 1978. 48 с.
4. Старова О.А. Розы Гвидо Гранди // Математика. Все для учителя. 2015. № 10(58). С. 30–32.

Реализация полиномиальной и сплайн интерполяции на Python

Г.И. Захарычев

Саровский физико-технический институт СарФТИ НИЯУ МИФИ

Данная статья описывает разработку программного средства для реализации полиномиальной и сплайн-интерполяции на основе данных от пользователя. Программа создана на языке программирования Python, с использованием подмодуля `scipy.interpolate`. Результаты данного исследования могут быть полезны для улучшений в учебном процессе, в частности для более широкого представления графической составляющей вычислительной математики.

Ключевые слова: вычислительная математика, интерполяция, функция, численный анализ, программное средство.

1. Введение

Интерполяция, интерполирование (от лат. Inter-polis – «разглаженный, подновлённый, обновлённый; преобразованный») – в вычислительной математике нахождение неизвестных промежуточных значений некоторой функции, по имеющемуся дискретному набору её известных значений, определенным способом. Термин «интерполяция» впервые употребил Джон Валлис в своём трактате «Арифметика бесконечных».

В функциональном анализе интерполяция линейных операторов представляет собой раздел, рассматривающий банаховы пространства как элементы некоторой категории.

Многим из тех, кто сталкивается с научными и инженерными расчётами, часто приходится оперировать наборами значений, полученных опытным путём или методом случайной выборки. Как правило, на основании этих наборов требуется построить функцию, на которую могли бы с высокой точностью попадать другие получаемые значения. Такая задача называется аппроксимацией. Интерполяцией называют такую разновидность аппроксимации, при которой кривая построенной функции проходит точно через имеющиеся точки данных.

2. Интерполяция

Задача интерполирования состоит в том, чтобы по значениям функции $f(x)$ в нескольких точках отрезка восстановить ее значения в остальных точках этого отрезка. Разумеется, такая задача допускает сколь угодно много решений. Задача интерполирования возникает, например, в том случае, когда известны результаты измерения $y_h = f(x, k)$ некоторой физической величины $f(x)$ в точках $x_k, k = 0, 1, \dots, n$, и требуется определить ее значения в других точках.

Интерполирование используется также при сгущении таблиц, когда вычисление значений $f(x)$ трудоемко. Иногда возникает необходимость приближенной замены или аппроксимации данной функции другими функциями, которые легче вычислить. В частности, рассматривается задача о наилучшем приближении в нормированном пространстве H , когда заданную функцию $f \in H$ требуется заменить линейной комбинацией ϕ заданных элементов из H так, чтобы отклонение $\|f - \phi\|$ было минимальным. Результаты и методы теории интерполирования и приближения функций нашли широкое применение в численном анализе, например, при выводе формул численного дифференцирования и интегрирования, при построении сеточных аналогов задач математической физики.

Для того чтобы рассмотреть интерполяцию, сгенерируем данные. Т.е. сгенерируем массив узлов интерполяционной сетки `x_data` и значений `y_data` некоторой функции f в узлах этой сетки при разном количестве известных точек.

Подмодуль `scipy.interpolate` содержит в себе методы для интерполяции.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy import interpolate
4 L = 0
5 R = np.pi
6 n_points = (4, 10, 64, 65)
7 f = np.sin
8 xh = np.linspace(L, R, num=100)
9 yh = f(xh)
10
11 3 usages
12 def generate_data(f, L, R, n):
13     x_data = np.linspace(L, R, n - 1)
14     y_data = f(x_data)
15     return x_data, y_data
16
17 4 usages
18 def plot_problem(ax, x_data, y_data):
19     ax.plot(xh, yh, linewidth=0.75, label="Исходная зависимость")
20     ax.scatter(x_data, y_data, marker="x", color="red", label="Известные данные")
21     ax.set_xlabel("$x$")
22     ax.set_ylabel("$y$")
23
24 fig, axs = plt.subplots(ncols=2, nrows=2, figsize=(16, 16), layout="tight")
25 for ax, N in zip(axs.flatten(), n_points):
26     x_data, y_data = generate_data(f, L, R, N)
27     plot_problem(ax, x_data, y_data)
28     ax.set_title(f"Исходные данные ${N}$")
29     ax.legend()
30
31 fig, axs = plt.subplots(ncols=2, nrows=2, figsize=(16, 16), layout="tight")
32 for ax, N in zip(axs.flatten(), n_points):
33     x_data, y_data = generate_data(f, L, R, N)
34     plot_problem(ax, x_data, y_data)
35     poly = interpolate.KroghInterpolator(x_data, y_data)
36     y_poly = poly(xh)
37     ax.plot(xh, y_poly, label="Интерполирующий полином")
38     ax.set_title(f"Результат интерполяции ${N}$")
39     ax.legend()
40
41 x_data, y_data = generate_data(f, L, R, n=5)
42 linear = interpolate.interpld(x_data, y_data, kind="linear")
43 y_linear = linear(xh)
44 fig, ax = plt.subplots(figsize=(10, 8), layout="tight")
45 ax.plot(xh, y_linear, label="Сплайн первой степени")
46 plot_problem(ax, x_data, y_data)
47 ax.legend()
48 ax.set_title("Кусочно-линейная интерполяция")
49 #Text(0.5, 1.0, 'Кусочно-линейная интерполяция')
50 cubic = interpolate.interpld(x_data, y_data, kind="cubic")
51 y_cubic = cubic(xh)
52 fig, ax = plt.subplots(figsize=(10, 8), layout="tight")
53 ax.plot(xh, y_cubic, label="Кубический сплайн")
54 plot_problem(ax, x_data, y_data)
55 ax.legend()
56 ax.set_title("Интерполяция кубическим сплайном")
57 #Text(0.5, 1.0, 'Интерполяция кубическим сплайном')
58 plt.show()
59
60
61
62

```

Рис. 1. Код программы

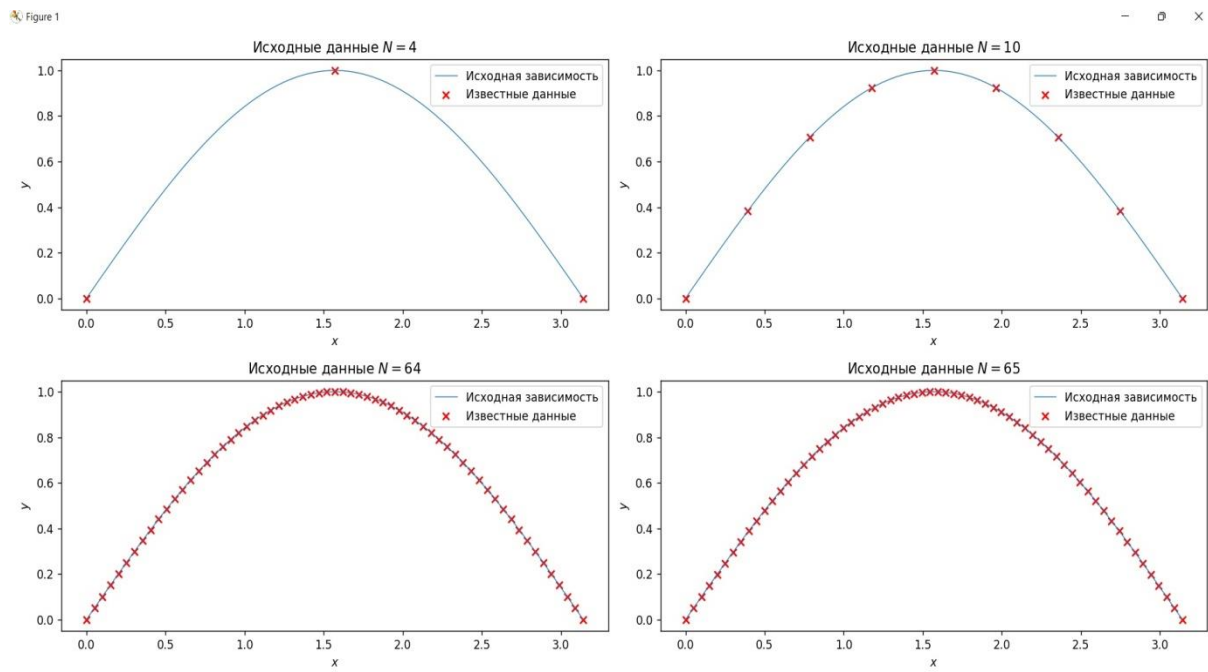


Рис. 2. Графики с исходными данными

Теперь когда данные готовы, осталось произвести интерполяцию. Методов интерполяции существует очень много. Рассмотрим полиномиальную интерполяцию и сплайн интерполяцию.

3. Полиномиальная интерполяция

Полиномиальная интерполяция основывается на том факте, что существует единственный полином $P_N(x) = a_0 + a_1x + \dots + a_Nx^N$ степени N , проходящий через точки $\{(x_1, y_1), \dots, (x_{N+1}, y_{N+1})\}$, если $x_i \neq x_j \forall i, j = 1, \dots, N+1, i \neq j$.

Недостатком полиномиальной интерполяции является её нестабильность. Полиномиальная интерполяция проявляет себя очень плохо при больших N .

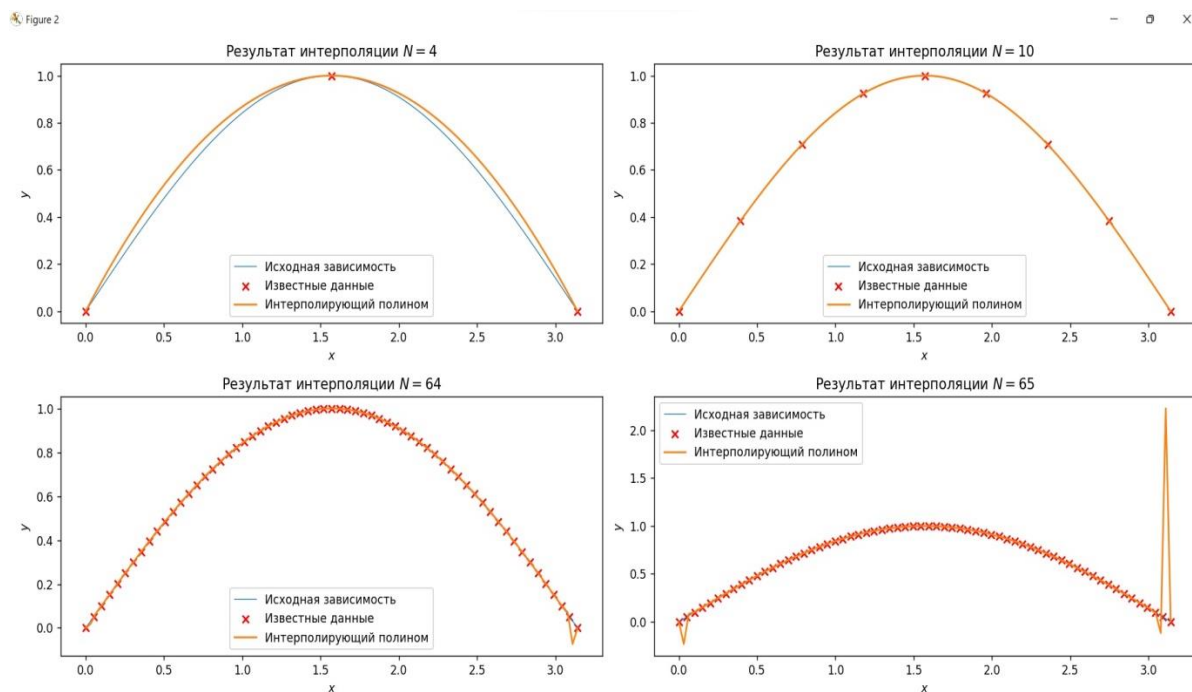


Рис. 3. Полиномиальная интерполяция

4. Сплайн интерполяция

Интерполирование многочленом ... на всем отрезке $[a, b]$ с использованием большого числа узлов интерполяции часто приводит к плохому приближению, что объясняется сильным накоплением погрешностей в процессе вычислений. Кроме того, из-за расходимости процесса интерполяции увеличение числа узлов не обязательно приводит к повышению точности. Для того чтобы избежать больших погрешностей, весь отрезок $[a, b]$ разбивают на частичные отрезки и на каждом из частичных отрезков приближенно заменяют функцию $f(x)$ многочленом невысокой степени (так называемая кусочно-полиномиальная интерполяция).

Одним из способов интерполирования на всем отрезке является интерполирование с помощью сплайн-функций. Сплайн-функцией или сплайном называют кусочно-полиномиальную функцию, определенную на отрезке $[a, b]$ и имеющую на этом отрезке некоторое число непрерывных производных.

Например, если взять в качестве сплайн-функции полином первой степени, то на выходе получится кусочно-линейная интерполяция.

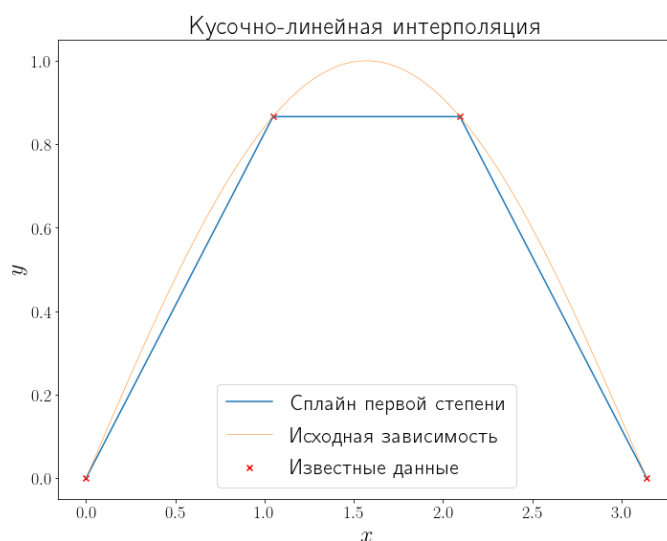


Рис. 4. Кусочно-линейная интерполяция

Очень часто выбирают полином третьей степени, которые позволяют строить гладкие интерполирующие полиному и разумно ведут себя между точками во многих случаях.

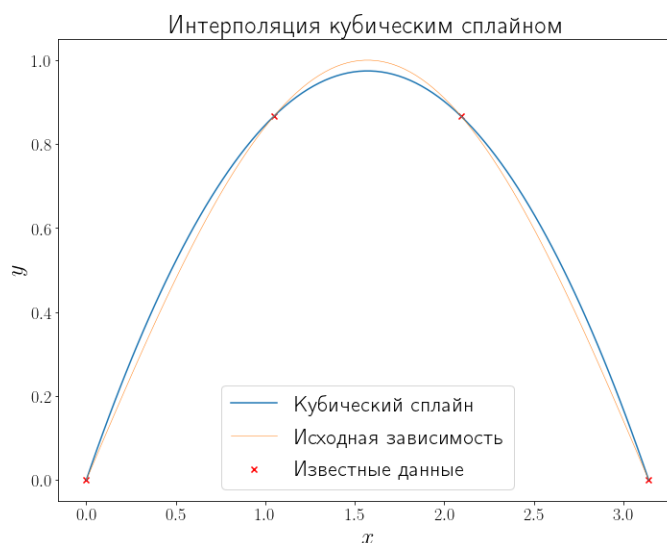


Рис. 5. Интерполяция кубическим сплайном

5. Заключение

В данной статье было рассмотрено использование подмодуля `scipy.interpolate` для реализации полиномиальной и сплайн-интерполяции на основе данных от пользователя. Это открывает большие возможности для визуализации математических функций и более детального рассмотрения интерполяции в целом. Реализованное решение может быть использовано в учебном процессе для более широкого представления графической составляющей вычислительной математики.

Разработанная программа так же может быть улучшена, путем добавления чтения данных из файла для более удобного взаимодействия пользователя с кодом.

Литература

1. Половко А.М. Интерполяция. Методы и компьютерные технологии их реализации, 2004.
2. Лутц М.А. Изучаем Python, 2009.
3. Картер Д. Python Библиотеки, 2024.
4. Дэн Б. Чистый Python. Тонкости программирования для профи, 2018.

Interpolation (`scipy.interpolate`) – Numpy and Scipy Documentation SciPy v1.13.0 Manual URL:
<https://docs.scipy.org/doc/scipy/reference/interpolate.html> (дата обращения: 04.04.2024).

Распределённая версия алгоритма глобального поиска в фреймворке интеллектуальной оптимизации «iOpt»

О.А. Зорин, Е.А. Козин

Нижегородский государственный университет им. Н.И. Лобачевского

При решении многих оптимизационных задач характерно, что время проведения очередного испытания может различаться. Это обусловлено тем, что при некоторых сочетаниях оптимизируемых параметров вычисления могут быть более трудоёмки. Как следствие, возникает проблема неэффективного использования вычислительных ресурсов. Организация параллельных вычислений в виде распределённой схемы позволяет с одной стороны задействовать несколько вычислительных узлов, с другой стороны равномерно распределить нагрузку между процессами, повышая общую эффективность процесса оптимизации.

Ключевые слова: глобальная оптимизация, многоэкстремальные задачи, параллельные вычисления.

1. Введение

В современном мире задачи поиска оптимальных решений встречаются практически во всех областях деятельности человека. Эти задачи часто обладают высокой сложностью, требуя системного подхода и использования специализированных методов для их решения. Нахождение оптимальных решений становится особенно сложным в условиях большого количества вариантов, противоречивых требований и изменяющегося представления об оптимальности решения.

Для решения таких задач широко используются алгоритмы глобального поиска. Алгоритмы глобального поиска можно разделить на детерминированные [1-4] и недетерминированные [5-9]. В данном исследовании предметом рассмотрения является известный алгоритм глобального поиска, предложенный Р. Г. Стронгиным [1,2,10-12]. Реализация данного алгоритма доступна в рамках фреймворка «iOpt» в открытом доступе [13].

Учитывая увеличение размерности пространства поиска и возрастающую сложность функций в современных задачах оптимизации, необходимость в интеграции параллельных вычислений и асинхронных методов в рамках рассматриваемого алгоритма становится особенно актуальной [10-12].

Использование параллельных вычислений позволяет распределить вычислительную нагрузку между несколькими процессорами или узлами, ускоряя выполнение вычислений и повышая общую производительность. Асинхронность позволяет выполнять вычисления без блокировки основного потока выполнения, что особенно важно при работе с распределёнными системами.

Таким образом, в данной статье предлагается исследовать эффективность использования вычислительных ресурсов при внедрении параллельных и асинхронных вычислений в рамках распределённой схемы глобального поиска.

В экспериментах для выполнения параллельных вычислений будут порождаться процессы Python, а в качестве эффективного механизма обмена информацией между ними будет использована база данных. Разработанная в рамках работы распределённая версия алгоритма глобального поиска будет интегрирована в фреймворк «iOpt».

2. Постановка задачи

Пусть имеется множество области поиска, заданное линейными ограничениями

$$D = \{ y \in \mathbb{R}^N : a_i \leq y_i \leq b_i, 1 \leq i \leq N \}, \quad (1)$$

где $y = (y_1, y_2, \dots, y_N)$ и $a_i, b_i \in \mathbb{R}$.

На этом множестве определена функция $f(y), y \in D$, которая удовлетворяет условию Липшица с коэффициентом L

$$\exists L > 0 \quad \forall y', y'' \in D \quad |f(y') - f(y'')| < L \|y' - y''\|. \quad (2)$$

Поставлена задача глобальной оптимизации, т.е. определение глобального минимума данной функции на заданном множестве

$$\min_{y \in D} f(y). \quad (3)$$

3. Метод решения

Вначале приведём базовый алгоритм глобального поиска [1]. Далее на его основе рассмотрим распределённую версию.

3.1. Базовый алгоритм глобального поиска

Пусть $y(x), x \in [0, 1]$, есть кривая Пеано. Тогда из непрерывности $f(y)$ и $y(x)$ и равенства $D = \{y(x) \mid 0 \leq x \leq 1\}$ следует, что $\min_{y \in D} f(y) = \min_{x \in [0, 1]} f(y(x))$.

Первые два испытания проводим на концах отрезка $x \in [0, 1]$, т.е. в точках $x^0 = 0$ и $x^1 = 1$. Определяем образы этих двух точек с помощью развёртки Пеано $y(x)$, т.е. получим $y^0 = y(0)$ и $y^1 = y(1)$. Также вычисляем соответствующие значения функции – $z^0 = f(y^0)$ и $z^1 = f(y^1)$. Количеству испытаний k выставляем значение равное единице.

Пусть проведено k испытаний, следующее испытание $k + 1$ проводится следующим образом.

1. Отсортировать вычисленные точки испытания в порядке возрастания их координат x , согласовать нижнюю индексацию с вычисленными образами y и значениями функции z :

$$0 = x_0 < x_1 < \dots < x_{k-1} < x_k = 1. \quad (4)$$

2. Принять в качестве приближённого оптимального значения:

$$z^* = \min\{z_i = f(y_i) \mid i = \overline{1, k}\}. \quad (5)$$

3. Вычислить оценки для неизвестной константы:

$$\mu = \max\left\{\frac{|z_i - z_{i-1}|}{(x_i - x_{i-1})^{1/N}} \mid i = \overline{1, k}\right\}. \quad (6)$$

4. Для каждого интервала (x_{i-1}, x_i) $i = \overline{1, k}$ вычислить характеристику:

$$R(i) = (x_i - x_{i-1})^{1/N} + \frac{(z_i - z_{i-1})^2}{r^2 \mu^2 (x_i - x_{i-1})^{1/N}} - 2 \frac{z_i + z_{i-1} - 2z^*}{r \mu}, \quad (7)$$

где $r > 1$ – параметр метода.

5. Найти номер t интервала с максимальным значением характеристики:

$$R(t) = \max\{R(i) \mid i = \overline{1, k}\}. \quad (8)$$

6. Провести очередное испытание в точке из интервала (x_{t-1}, x_t) :

$$x^{k+1} = \frac{x_t + x_{t-1}}{2} - \frac{\text{sign}(z_t - z_{t-1})}{2r} \left(\frac{|z_t - z_{t-1}|}{\mu} \right)^N, \quad (9)$$

вычислить образ новой точки $y^{k+1} = y(x^{k+1})$, значение функции $z^{k+1} = f(y^{k+1})$ и увеличить номер шага поиска на единицу: $k = k + 1$.

Остановка алгоритма происходит, если выполнено следующее условие:

$$(x_t - x_{t-1})^{1/N} \leq \varepsilon, \quad (10)$$

где ε – заданная по координатной точности задачи. Либо по достижению заданного максимального количества испытаний.

В качестве оптимального берётся значение:

$$z^* = \min\{z^i \mid i = \overline{1, k}\}. \quad (11)$$

3.2. Распределённая версия

В распределённой схеме выделяется один из имеющихся процессов в роли главного. Его основная задача заключается в управлении и контроле выполнения оптимизационного процесса. Он обеспечивает хранение поисковой информации о проведённых испытаниях, осуществляет вычисление и передачу точек для следующих испытаний свободным рабочим процессам, а также осуществляет сбор и обработку полученных результатов. Рабочие процессы, в свою очередь, выполняют лишь несколько основных операций: получение точек испытания от главного процесса, вычисление значения целевой функции в данных точках и передачу результатов обратно главному процессу. В соответствии с этим правила проведения испытаний в базовом алгоритме подвергаются изменениям (см. рис. 1).

Допустим, имеется p рабочих процессов. Первые p испытаний проводятся равномерным образом. То есть на основном процессе точки равномерно распределяются по отрезку $x \in [0, 1]$, определяются их образы, далее они рассылаются рабочим процессам, где подсчитывается их значения функции, после чего возвращаются обратно на основной процесс. Начальный шаг – блокирующий, для корректного выполнения следующих испытаний.

Когда все исполняющие процессы заняты, алгоритм находится в состоянии простоя. Если на определенном шаге освободилось $q \leq p$ процессов, в базовый алгоритм добавляются следующие действия.

Перед выполнением первого правила собирается информация с освободившихся процессов. То есть к поисковой информации добавляются принятые точки испытаний x^j с их образами $y(x^j)$ и вычисленными значениями функции $f(y(x^j))$ ($j = \overline{1, q}$). Количество испытаний k становится равным $k + q$.

Правило 5 и 6 заменяются на

5. Характеристики $R(i), i = \overline{1, k}$ отсортировать в порядке невозрастания:
$$R(t_1) \geq R(t_2) \geq \dots \geq R(t_{k-1}) \geq R(t_k). \quad (12)$$

Выбрать из них q штук наибольших характеристик с номерами $t_j, j = \overline{1, q}$.

6. В соответствующих интервалах $(x_{t_{j-1}}, x_{t_j}), j = \overline{1, q}$ провести очередные испытания по формуле (7), вычислить образы новых точек $y^{k+j} = y(x^{k+j})$ и передать их освободившимся рабочим процессам.

3.3. Реализация в фреймворке «iOpt»

Распределённая версия алгоритма глобального поиска интегрируется в фреймворк «iOpt» как один из модулей, предоставляющих пользователю возможность выбора оптимального метода решения в зависимости от характеристик задачи.

В качестве механизма обмена данными используется база данных, где хранится информация о точках испытаний, их образах и значениях функции. Главный процесс и рабочие процессы взаимодействуют с базой данных для получения и передачи данных, обеспечивая целостность и доступность информации. Процесс взаимодействия отражён на блок-схеме (см. рис. 1).

База данных в качестве механизма обмена данных позволяет эффективно обмениваться информацией между процессами, порождёнными на различных вычислительных узлах. Этот подход выбран из-за ограничений стандартных средств управления процессами в Python, которые позволяют порождать процессы только в рамках одного вычислительного узла.

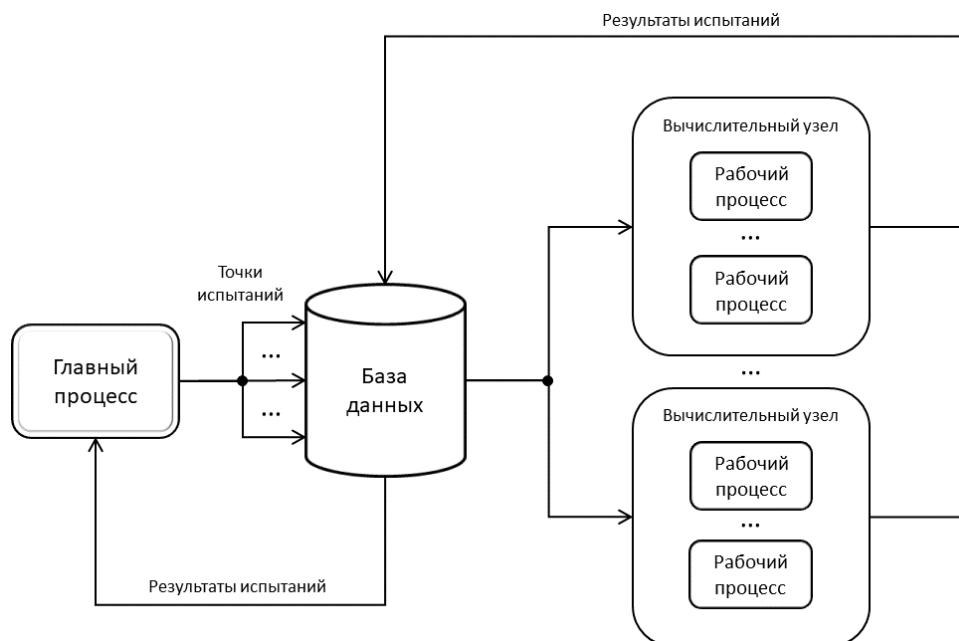


Рис. 1. Блок-схема распределённой схемы

4. Численные эксперименты

4.1. Задача коммивояжёра

В качестве критерия использовалась задача коммивояжёра (Travelling Salesman Problem, TSP), которая представляет собой классическую задачу комбинаторной оптимизации, где необходимо найти самый короткий маршрут для коммивояжёра, проходящий через каждый город из заданного списка ровно один раз и возвращающийся в начальный город. В математической формулировке это означает поиск такой перестановки городов, которая минимизирует общее расстояние путешествия. Эта задача имеет важное значение в теоретической информатике и исследованиях операций, так как является NP-трудной.

Для решения задачи используются различные алгоритмы, включая метод ближайшего соседа, минимального остовного дерева, ветвей и границ, а также генетические алгоритмы. В данной работе исследуется применение генетического алгоритма для решения задачи коммивояжёра с использованием фреймворка `scikit-opt` [14]. Задача состоит в настройке параметров генетического алгоритма для получения оптимального решения в рамках ограниченного количества итераций. В качестве входных данных для алгоритма используется матрица расстояний между городами, взятая из известной базы данных TSPLIB [15].

4.2. Результаты

Численные эксперименты проводились с использованием ресурсов СК «Лобачевский». Для алгоритма глобального поиска параметр надёжности метода из (7) и (9) был выбран $r = 4$. Остановка метода производилась по достижению максимального числа испытаний, заданным 200. В качестве базы данных использовалась PostgreSQL. Для более корректного измерения времени вычисления решение задачи было выполнено 10 раз с последующим усреднением результата.

Результаты численных экспериментов отражены в таблице 1. Из представленных данных можно сделать вывод, что увеличение количества узлов и рабочих процессов приводит к значительному сокращению времени вычислений. При этом точность решения задачи остаётся на том же уровне.

Стоит отметить, что при решении задач, не требующих значительных вычислительных ресурсов, применение распределенной версии алгоритма может привести к ухудшению производительности из-за превышения накладных расходов. Это связано с дополнительной сложностью в организации взаимодействия между процессами и передачей данных. Для достижения оптимальной производительности в таком случае рекомендуется обратить внимание на асинхронную схему, реализованную в фреймворке «iOpt».

Таблица 1. Среднее время вычисления и результат решения задачи коммивояжёра

Количество узлов	Число рабочих процессов (на одном узле)	Результат (меньше – лучше)	Время вычисления (сек.)	Ускорение
1	1	16 315.7	396.1	1
	2	16 314.3	197.5	2.0
	5	16 206.6	80.7	4.9
	10	16 301.4	44.4	8.9
	20	16 260.5	27.9	14.2
2	5	16 238.0	43.4	9.1
	10	16 268.4	24.6	16.1
	20	16 194.7	16.8	23.6
3	5	16 179.4	30.3	13.1
	10	16 257.5	18.4	21.6
	20	16 595.7	12.6	31.4
4	5	16 302.1	24.2	16.4
	10	16 287.9	15.3	25.9
	20	16 435.9	10.8	36.7

5. Заключение

В рамках научно-исследовательской работы была предложена новая схема параллельных вычислений для алгоритма глобального поиска – распределенная схема. Анализ результатов численных экспериментов явно показал, что увеличение числа рабочих процессов на нескольких вычислительных узлах существенно сокращает время выполнения алгоритма. Этот факт подтверждает эффективность предложенного подхода в решении сложных задач оптимизации.

Исследование также указывает на значимость такого подхода для разработки более рациональных стратегий оптимизации и распределения вычислительных ресурсов. Понимание влияния параллельных вычислений на производительность алгоритмов позволяет оптимизировать работу системы и добиваться лучших результатов при решении сложных оптимизационных задач.

Эти результаты могут служить основой для дальнейших исследований в области оптимизации вычислительных процессов и распределения ресурсов. Они могут быть полезны для разработки более эффективных алгоритмов оптимизации и стратегий управления вычислительными системами.

Литература

1. Стронгин Р.Г. Численные методы в многоэкстремальных задачах (информационно-статистические алгоритмы) – М.: Главная редакция физико-математической литературы издательства "Наука", 1978. – 240 с.

2. Strongin R.G., Sergeyev Y.D. Global Optimization with Non-Convex Constraints. Sequential and Parallel Algorithms. – The Netherlands, Dordrecht: Kluwer Academic Publishers, 2000. – 728 p.
3. Сергеев Я.Д., Квасов Д. Диагональные методы глобальной оптимизации. – ФИЗМАТЛИТ. – 2008. – 352 с.
4. Sergeyev Y.D., Strongin R.G., Lera D. Introduction to global optimization exploiting space filling curves. – Verlag New York:Springer. – 2013. – 125 p.
5. Yang, X.-S. Nature-inspired metaheuristic algorithms. – Frome: Luniver Press. – 2008. – 148 p.
6. Kennedy J., Eberhard R. Particle swarm optimization // Proc. of IEEE Intern. Conf. on Neural Networks. – 1995. – V. 4. – P. 1942–1948.
7. Zhigljavsky, A.A. Theory of Global Random Search. – Dordrecht: Kluwer Academic Publishers. – 1991. – 341 p.
8. Карпенко А.П., Селиверстов Е.Ю. Обзор методов роя частиц (particle swarm optimization) // Электронное научно-техн. издание “Наука и образование”. – 2009. – № 3.
9. Кошур, В.Д. Глобальная оптимизация на основе гибридного метода усреднения координат и метода роя частиц // Ж. Вычислительные технологии. – 2013. – №4(18) .– С. 36-47.
10. Стронгин Р.Г., Гергель В.П., Гришагин В.А., Баркалов К.А. Параллельные вычисления в задачах глобальной оптимизации. – М.: Издательство Московского университета. 2013. – 280 с.
11. Sergeyev Y.D., Grishagin V.A. A Parallel Method for Finding the Global Minimum of Univariate Functions // Journal of Optimization Theory and Applications. – 1994. – V. 80(3). – P. 513–536.
12. Sergeyev Y.D., Grishagin V.A. Parallel Asynchronous Global Search and the Nested Optimization Scheme // J. Comput. Anal. Appl. – 2001. – V. 3(2). – P. 123–145.
13. Официальный сайт с исходным кодом фреймворка iOpt. – Режим доступа: <https://github.com/aimclub/iOpt>. – Загл. с экрана.
14. Официальный сайт с исходным кодом библиотеки scikit-opt – Режим доступа: <https://scikit-opt.github.io>. – Загл. с экрана.
15. Официальный сайт TSPLIB. – Режим доступа: <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>. – Загл. с экрана.

Разработка обучающего интерактивного онлайн портала по дисциплине «Методы оптимизации»

П.А. Кочетков

Мордовский государственный университет им. Н.П. Огарёва

В современном мире, где каждый день появляются новые задачи, требующие оптимизации, умение применять методы оптимизации и использовать специальные инструменты для этого – ключевой навык для многих профессий. Создание интерактивного онлайн портала позволит более эффективно учиться и практиковаться студентам, использующим его.

Ключевые слова: онлайн портал, сайт, программирование, HTML, JavaScript, методы оптимизации.

1. Введение

Методы оптимизации — это важная и широко применяемая наука, начавшая активно развиваться в XVIII веке. Она занимается разработкой алгоритмов и методов для поиска оптимальных решений в различных областях знаний.

Оптимизация позволяет найти лучшие решения задачи при заданных ограничениях и условиях, что является важным для многих профессий и сфер деятельности. Методы оптимизации играют важную роль в решении многих реальных задач, улучшении качества жизни и развитии науки и технологий.

В нашем быстро меняющемся мире, где каждая новая задача требует поиска оптимального решения, владение методами оптимизации становится неотъемлемым навыком для специалистов самых разных областей. Внедрение инновационных технологий и автоматизация процессов требуют от профессионалов умения быстро адаптироваться и применять наиболее эффективные подходы к решению возникающих проблем.

Создание онлайн-платформы для изучения методов оптимизации открывает новые возможности для студентов, позволяя им приобретать необходимые навыки в удобном формате. Такой портал будет способствовать развитию критического мышления, аналитических способностей и умения работать с большим объемом информации.

Важно отметить, что применение методов оптимизации не ограничивается только техническими специальностями. Они также могут быть использованы в бизнесе, экономике, медицине, образовании и многих других сферах деятельности. Внедрение таких методов позволяет повысить эффективность работы, снизить затраты и улучшить качество конечного продукта или услуги.

Таким образом, создание онлайн-платформы для изучения методов оптимизации является важным шагом в подготовке квалифицированных специалистов, способных успешно справляться с вызовами современного мира.

2. Онлайн портал

Онлайн-портал — это веб-сайт, который предоставляет пользователям множество различных функций и услуг через интернет. Обычно онлайн-порталы содержат большое количество информации и предлагают различные интерактивные сервисы, такие как форумы, блоги, социальные сети, электронная почта, новости, погода, поиск и другие. Они могут быть ориентированы на определенную аудиторию (например, образовательный портал) или предлагать услуги для широкого круга пользователей.

Основное отличие онлайн-портала от обычного веб-сайта заключается в его масштабе и функциональности. Веб-сайт обычно представляет собой одну страницу или набор страниц,

которые предоставляют информацию о компании, продукте или услуге. В то же время онлайн-портал предлагает гораздо больше возможностей для взаимодействия с пользователями.

Существуют различные типы интернет-порталов. Некоторые из них специализируются на определенной тематике, например, финансовые, спортивные, развлекательные или образовательные порталы. Другие же являются универсальными, предлагая своим пользователям разнообразные сервисы и информацию.

В целом, интернет-порталы играют важную роль в современной интернет-культуре, предоставляя пользователям широкий спектр услуг и информации. Они стали незаменимым инструментом для бизнеса, образования, развлечений и многих других сфер жизни.

3. Структура портала

Рассмотрим случай из нашей предметной области, а именно образовательный портал. Обучающий интерактивный онлайн портал должен включать в себя несколько модулей направленных на обучение студентов. Их структура должна быть логичной и интуитивно понятной, чтобы пользователи могли легко перемещаться между разделами и находить нужную информацию. Пользовательский интерфейс должен быть максимально простым и понятным, чтобы пользователи могли быстро найти нужную информацию и использовать все возможности портала. Добавление системы тестирования позволит пользователям проверять свои знания и прогресс в обучении, а использование видео и аудио контента поможет сделать обучение более интересным и эффективным.

Исходя из этих особенностей, планируется, что сайт будет состоять из пяти модулей: онлайн курс, Калькулятор, видеоматериалы, история методов оптимизации, а также литература. Разберем подробнее каждый из них.

Модуль "Онлайн курс": Этот модуль должен включать в себя полный учебный план курса по методам оптимизации. Курс должен быть структурирован таким образом, чтобы студенты могли легко следовать ему и понимать материал. Каждый урок должен содержать теоретический материал, практические задания, тесты для самопроверки и примеры решения задач.

Модуль "История методов оптимизации": этот модуль должен содержать исторический обзор развития методов оптимизации. Студенты должны узнать о том, как развивались методы оптимизации, какие были ключевые моменты в их развитии, и какие ученые внесли наибольший вклад в эту область.

Модуль "Видеоматериалы": этот модуль может содержать видеолекции, презентации, демонстрационные материалы и другие мультимедийные ресурсы, которые помогут студентам лучше понять методы оптимизации.

Модуль "Калькулятор": этот модуль может содержать онлайн-калькуляторы, которые помогут студентам решать задачи по методам оптимизации. Например, калькулятор для нахождения экстремума функции или для решения системы линейных уравнений.

Модуль "Литература": этот модуль должен содержать список рекомендуемой литературы по методам оптимизации. Это могут быть учебники, научные статьи, монографии и другие источники, которые помогут студентам углубить свои знания в этой области.

4. Инструменты разработки

Чтобы реализовать данный проект, потребовались средства разработки. Прежде всего, это языки программирования: HTML, позволяющий определить структуру веб-сайта путем создания HTML-страниц и размещения на них необходимых элементов, а также установления ссылок на другие HTML-страницы, образующие сайт вместе. Язык разметки CSS, который дает возможность придать стиль созданным элементам HTML, обеспечивая таким образом дизайн сайта. JavaScript, благодаря которому работают анимации и интерактивная часть сайта

Для написания кода была использована интегрированная среда разработки Visual Studio Code, которая предлагает удобный интерфейс для написания кода на выбранных языках программирования и обладает множеством полезных функций, упрощающих процесс разработки.

5. Дизайн портала

Дизайн обучающего интерактивного онлайн портала по дисциплине «Методы оптимизации» должен быть современным, удобным и привлекательным для пользователей. Портал должен быть доступен и удобен для использования на любых устройствах, будь то компьютер, планшет или смартфон.

На главной странице содержится краткое описание каждого раздела, чтобы пользователи могли сразу понять, какой материал им нужен. Планируется создание удобной навигации по сайту, чтобы пользователи могли быстро найти нужный раздел или материал.

В ходе работы были разработаны макеты готового решения для главной страницы и для страниц с навигацией. Сверху находится так называемая «шапка» сайта, на которой слева расположена кнопка перехода на главный экран сайта. Снизу — «подвал» сайта, в котором находится информация об университете для пользователей, не обучающихся в нашем ВУЗе, а также ссылка на сайт университета.

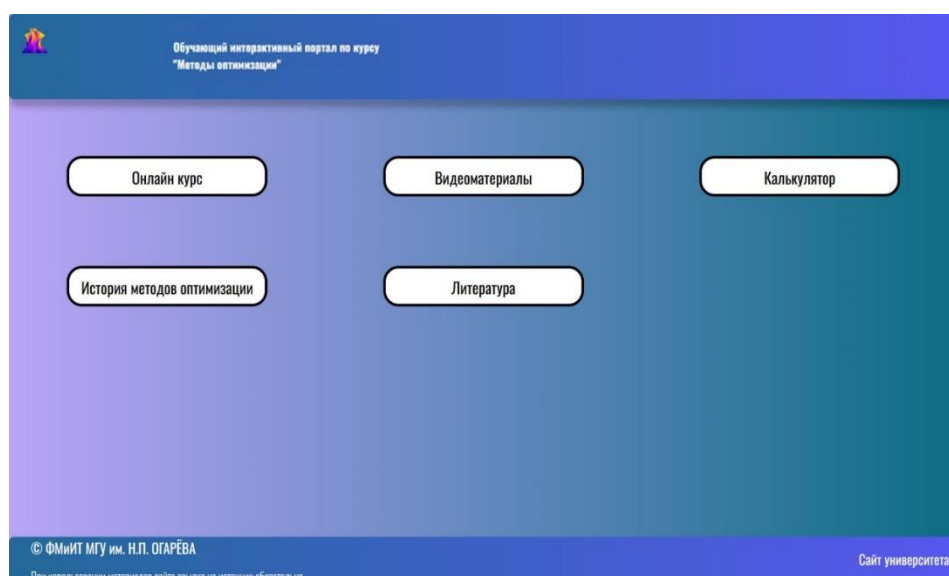


Рис. 1. Макет главной страницы онлайн портала

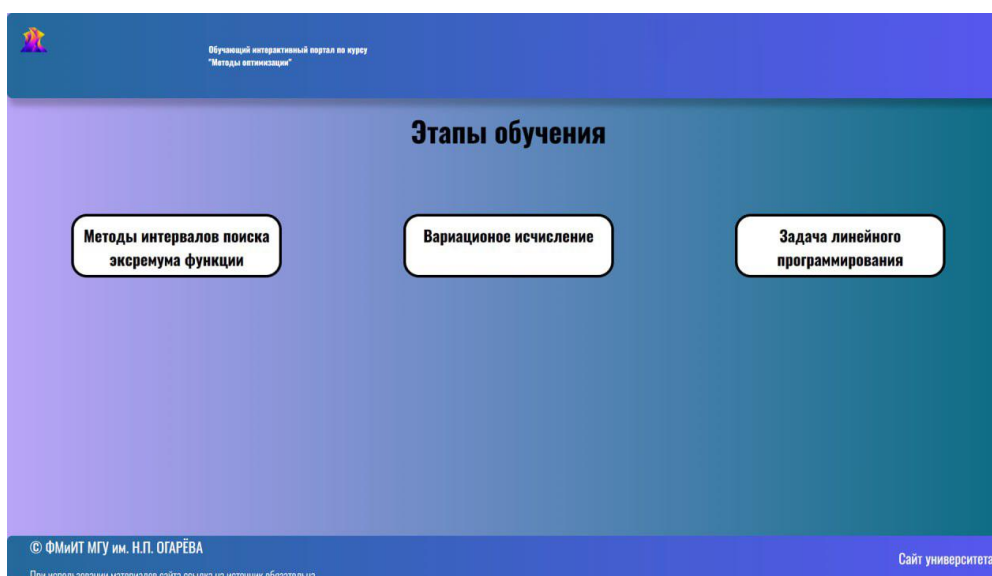


Рис. 2. Макет страницы навигации модуля “Онлайн курс”

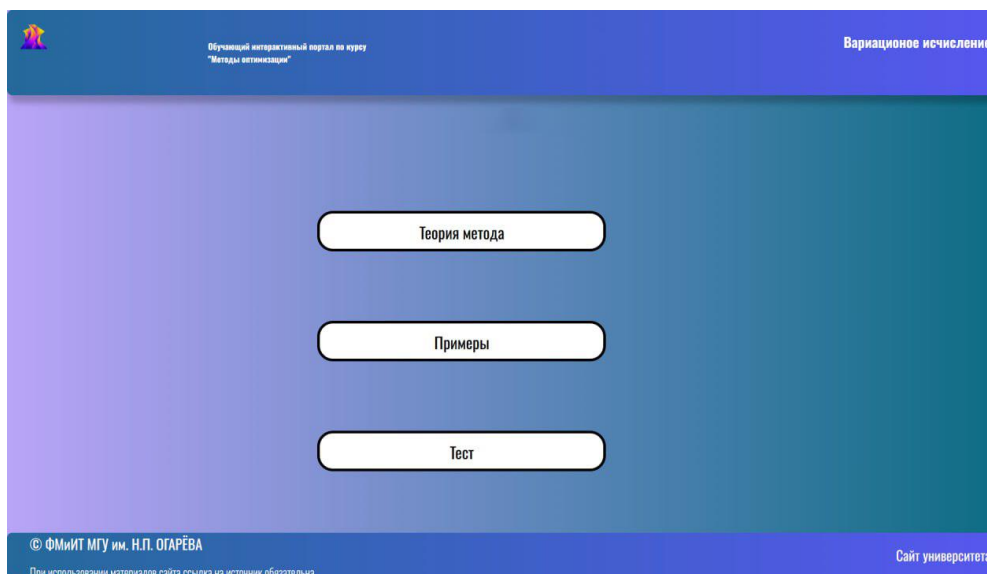


Рис. 3. Макет страницы навигации секции “Вариационное исчисление” модуля “Онлайн курс”

Литература

1. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. – 2-е изд. – Санкт-Петербург: Питер, 2014. – 320 с.
2. Флэнаган Д. JavaScript. Подробное руководство. – Пер. с англ. – СПб: Символ-Плюс, 2008. – 992 с.
3. Прокопенко Н.Ю. Методы Оптимизации: учебное пособие – Нижний Новгород: ННГАСУ, 2018. – 118 с.
4. Таха Х.А. Введение в исследование операций. – Москва: Вильямс, 2005. – 912 с.
5. Аттетков А.В., Зарубин В.С., Канатников А.Н. Введение в методы оптимизации – Москва: Финансы и статистика, 2008. – 272 с.
6. Базара М., Шетти К. Нелинейное программирование. Теория и алгоритмы. – Москва: Мир, 1982. – 583 с.
7. Пантелеев А. В., Летова Т.А. Методы оптимизации в примерах и задачах: учебное пособие. – 4-е изд., испр. – СПб.: Издательство "Лань", 2015. – 512 с.
8. Акулич И.Л. Математическое программирование в примерах и задачах. Минск: Высшая школа, 2009. – 98 с.

Диагностика сердечно-сосудистых заболеваний с использованием сверточной нейронной сети

В.В. Кузнецов, Н.Ю. Золотых

Нижегородский государственный университет им. Н.И. Лобачевского

В данной работе мы предлагаем метод диагностики сердечно-сосудистых заболеваний на основании данных электрокардиограмм (ЭКГ) с использованием аппарата нейронных сетей, а именно - сверточной нейронной сети. Нашей целью является построение модели, которая может определять большой набор сердечно-сосудистых заболеваний с использованием сигнала и сегментации, построенной на этом сигнале, с высокой степенью точности. Сигнал представляет собой волны с частотой дискретизации 100 и 500 Гц и длинами сигналов 10 секунд. Некоторые из диагнозов показывают хорошие метрики качества. Автоматическая диагностика сердечно-сосудистых заболеваний призвана сократить и оптимизировать работу врачей-кардиологов.

Ключевые слова: диагностика, сверточная нейронная сеть, ЭКГ, электрокардиография, сердечно-сосудистая система.

1. Введение

В современной медицине остро стоит вопрос определения и диагностики сердечно-сосудистых заболеваний. Большое число людей страдает подобными заболеваниями. Зачастую происходят ошибки в постановке диагнозов. В качестве помощи и корректировки работы кардиологов, мы хотим предложить автоматизированные методы определения патологий сердца.

Исследовать состояние пациента мы будем на основе ЭКГ – записи электрической активности сердца, получаемой за счет наложения электродов к различным частям тела человека. В последнее время автоматический анализ ЭКГ имеет большое значение. На каждом ЭКГ можно выделить 5 основных зубцов: P, Q, R, S, T, которые описывают удар сердца [1].

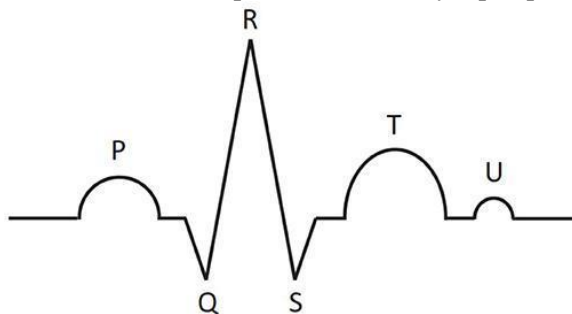


Рис. 1. Схематическое представление основных частей ЭКГ сигнала для одного кардиоцикла: P, T, U волны и QRS комплекс, включающий Q, R и S пики.

Кардиологи в своей работе используют специальные алгоритмы для определения ритмов, электрической оси сердца и всевозможных заболеваниях сердечно-сосудистой системы.

Мы же хотим не просто автоматизировать данные алгоритмы, а предложить абсолютно новый способ диагностики сердечных патологий с использованием аппарата нейронных сетей. Используя ЭКГ в виде последовательности значений сигнала, мы хотим, чтобы нейронные сети научились ставить диагнозы подобно тому, как это делают врачи.

Мы считаем, что данное направление достаточно перспективно, т.к. в случае решения поставленной задачи диагностирование заболеваний станет проще и быстрее. К тому же нейронные сети помогут в определении диагнозов, требующих больших финансовых и человеческих ресурсов, тем самым делая медицину более доступной для людей.

На данную тематику проводятся научные исследования, которые освещены в статье [2].

Работа [2] посвящена диагностике синусового ритма, шума (когда определение ритма невозможно) и 12 видов аритмий.

На вход данная нейронная сеть принимает последовательность, которая по своей сути представляет временной ряд. Длина последовательности составляет 30 сек. Частота дискретизации ЭКГ – 200 Гц. База данных в данном случае составляла 64 121 ЭКГ от 29 163 пациентов.

В данной статье определения диагнозов авторы используют 34-слойную сверточную нейронную сеть с 33 сверточными слоями и полносвязным слоем. В начале используются свертка, нормализация выхода и функция-активации ReLU. Далее сеть состоит из 16 блоков, в каждом из которых последовательно следуют свертка, нормализация, функция активации, регуляризация (Dropout) и еще одна свертка. В конце сети вновь происходит нормализация, используется функция-активации и далее следует выходной полносвязный слой, который дает 14 элементов на выходе.

В статье приведено описание результатов экспериментов, основанное на вычислении статистических показателей эффективности диагностики. Авторы использовали точность, чувствительность и F_1 -меру для оценки качества работы модели. Основной уклон делается на сравнение машинных результатов и выводов кардиологов. Показатели сравниваются как для случая с использованием последовательностей сигналов на входе, так и для случая, когда на вход подается вектор уникальных аритмий, которые есть в 30 записях ЭКГ. В каждом из показателей машинные методы оказались лучше. Отметим статистические данные, полученные машинными методами в Таблице 1.

Таблица 1. Значения метрик в рассматриваемой статье.

Метрика	Значение метрики
<i>Precision</i>	0.8
<i>Recall</i>	0.784
$F_1 - score$	0.776
<i>Precision</i> (вектор)	0.809
<i>Recall</i> (вектор)	0.827
$F_1 - score$ (вектор)	0.809

Одна из задач, которую мы ставим – улучшение результатов, полученных в описанной выше работе.

2. Алгоритм

2.1. Предварительная подготовка

Исходные данные представляют собой записи ЭКГ от 18885 пациентов [3]. Они также содержат диагнозы по каждой болезни. Важно отметить, что эти записи содержат данные по каждому из 12 отведений. Извлекая записи, получаем, что наша выборка представляет собой 21837 ЭКГ с 12 отведениями и длительностью 10 с, которые приводятся к частоте дискретизации 500 Гц.

Предобработка заключается в нормализации исходных данных. Для этого мы находим среднее арифметическое сигналов по каждому отведению. Находим разность между исходными значениями сигнала и их средним и делим полученный результат на стандартное отклонение по каждому отведению.

2.2. Архитектура нейронной сети

У нашей сверточной нейронной сети имеется 2 входа.

Первый вход представляет собой последовательность размером $(l, 12)$, где l – длина, поступающего на вход сигнала, а 12 – число отведений.

Второй вход – это выход с другой нейронной сети, задача которой – сегментация ЭКГ [4], имеет размер $(l, 4)$, где 4 – величина, отражающая принадлежность данного значения сигнала к сегментам P, QRS, T или ни одному из вышеперечисленных.

Конкретно в нашем случае значение l мы выбрали 4500, т.е. 9 секунднй сигнал с частотой дискретизации 500 Гц.

Далее, от первых двух входов следуют 2 блока, в каждом из которых присутствует сверточный слой, при помощи которого мы стремимся выявить некоторые признаки и свойства в наших последовательностях сигналов и сегментаций. Число фильтров, используемых в каждом из сверточных слоев различно.

Каждый сверточный слой имеет одинаковый параметр размера ядра: 9 (размер ядра – это величина, определяющая длину окна свертки). Число фильтров в слоях удобно записать в виде списка. Для входа с сигналами по 12 отведений этот список имеет вид:

$filters = [32, 32]$, где $filters$ отражает размерность выходного пространства (количество выходных фильтров в свертке).

Для входа с принадлежностью к тому или иному сегменту ЭКГ (сегментации) этот список имеет вид: $filters = [32, 32]$.

Получаем тензоры выходных данных для каждого фильтра. Глубина набора тензоров составит число фильтров.

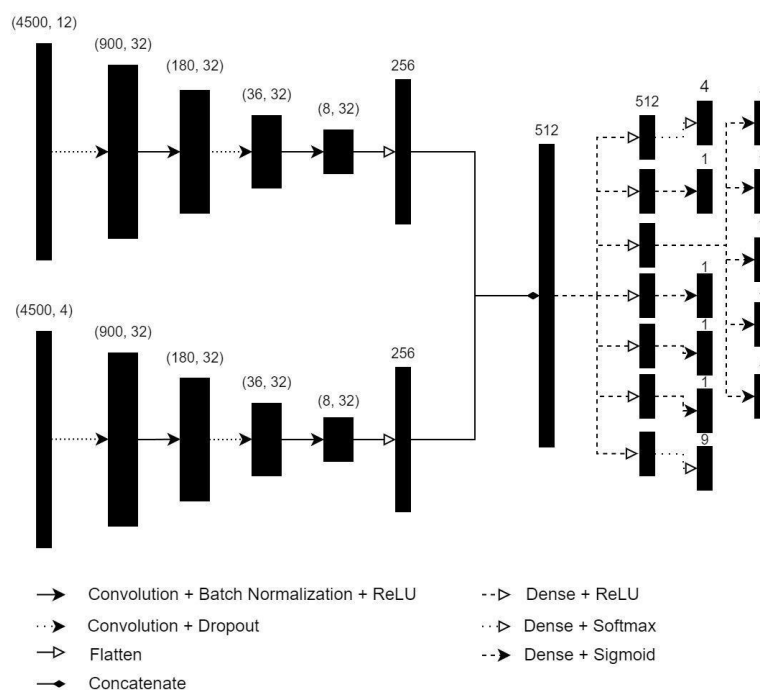


Рис. 2. Архитектура сверточной нейронной сети.

Далее следует Dropout – основной инструмент для борьбы с переобучением в нашей сети. Dropout выбрасывает (скрывает) определенный процент нейронов, которые отсутствуют в дальнейших вычислениях. Более обученные нейроны получают больший вес. В нашем случае величина отбрасываемых нейронов составляет 0.25 от числа всех имеющихся.

После этого следует очередной сверточный слой с количеством фильтров из $filters$. Далее в каждом из 2 блоков после сверточного слоя присутствует слой с нормализацией, идея которой заключается в том, что сеть обучается по батчам, то есть ошибка считается по какому-то подмножеству обучающей выборки. Данный способ нормировки подразумевает, что мы рассматриваем некоторое распределение элементов батча, вычитаем из них среднее и делим на дисперсию выборки, получив распределение с центром в 0 и дисперсией 1. Такое распределение позволяет сети быстрее обучаться, т.к. все числа получаются одного порядка и, соответственно, в заданном диапазоне. В дальнейшем, для получения выхода со слоя используются 2 параметра, отвечающие за сжатие и сдвиг [5].

И последний элемент каждого из 2 блоков: функция активации ReLU. Это своего рода функция-выпрямитель.

А после этих блоков следует слой Flatten, сводящий полученную нами последовательность к вектору. Далее производится конкатенация полученных векторов.

Затем происходит разветвление в зависимости от поставленной задачи. В нашем случае от конкатенации двух векторов у нас следует 6 ответвлений для определения:

1. Синусового ритма, фибрилляций, трепетаний или другого несинусового ритма;
2. Наличие или отсутствие любой гипертрофии;
3. Наличие или отсутствие любой из экстрасистол;
4. Наличие или отсутствие любой из АВ-блокады;
5. Наличие или отсутствие блокады ножек пучка Гиса;
6. Определение электрической оси сердца.

В каждом ответвлении следуют 2 полносвязных слоя с определенными весами. Первый слой дает нам на выходе 512 нейронов с функцией активации ReLU, а последний обычный плотный связанный слой Dense позволяет нам организовать вывод определенного числа нейронов с учетом функции активации. Итоговый результат формируется на основе ядра – матрицы весов и вектора смещения.

Функции активации в данном случае на последнем слое может отличаться. Если на выходе 1 нейрон, то используем Sigmoid, если больше (4 для определения ритмов, 9 для электрических осей), то Softmax [6].

Sigmoid – это логистическая функция, определяемая следующим образом:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Softmax – это обобщение логистической функции для многомерного случая. Функция преобразует вектор x заданной размерности в вектор той же размерности, где каждая координата полученного вектора представлена вещественным числом в интервале $[0;1]$, и сумма координат равна 1. Данная функция имеет вид:

$$f(x)_i = \frac{e^{x_i}}{\sum_{k=1}^K e^{x_k}}$$

Всего сеть имеет 11 выходов по определению того или иного диагноза. Перечислим их:

1. Наличие синусового ритма, фибрилляций, трепетаний или другого несинусового ритма;
2. Наличие или отсутствие любой гипертрофии;
3. Наличие или отсутствие любой из экстрасистол;
4. Наличие или отсутствие любой из АВ-блокады;
5. Наличие или отсутствие блокады правой ножки пучка Гиса, неполной;
6. Наличие или отсутствие блокады левой ножки пучка Гиса, неполной;
7. Наличие или отсутствие блокады задней ветви левой ножки пучка Гиса;
8. Наличие или отсутствие блокады передней ветви левой ножки пучка Гиса;
9. Наличие или отсутствие блокады правой ножки пучка Гиса, полной;
10. Наличие или отсутствие блокады левой ножки пучка Гиса, полной;
11. Определение электрической оси сердца.

У первого выхода для ритмов 4 нейрона, у последнего для электрических осей 9 нейронов. В остальных случаях на выходе 1 нейрон.

3. Эксперименты

3.1. Подготовка данных для обучения

Исходный набор данных состоит из 21837 ЭКГ по 12 отведений с длительностью 10 секунд. Для каждой ЭКГ присутствует разметка, отражающая принадлежность каждого сигнала к сегментам P, QRS, T или ни одному из перечисленных [4]. После предварительной обработки

данных, описанной выше, делим нашу общую выборку на обучающую и тестовую в процентном соотношении 70 на 30.

Для предотвращения переобучения была использована аугментация [7]. На вход нейронной сети мы будем подавать не весь 10 секундный сигнал, а лишь его часть и делать на основе этой части выводы. Будем «вырезать» 9 секундные сигналы из исходных данных. Для определения намеченных нами диагнозов данные обрезания ЭКГ допустимы. На 9 секундах результат будет аналогичен результату на 10 секундах. Тем самым мы расширяем максимально допустимое множество входных последовательностей. Отсюда и переобучиться нам будет достаточно проблематично.

Добавим, что многие задачи имеют крупный дисбаланс между классами. Поэтому наша задача раздать веса классам, чтобы их «уравнять». Делаем это при помощи пропорционального деления общего количества экземпляров на количество классов и количество представителей конкретного класса, для которого рассчитывается вес.

3.2. Метрики качества

Необходимо определить нулевую гипотезу. В нашем случае это: «Наличие у пациента некоторого (в зависимости от задачи) заболевания». В случае бинарной задачи мы принимаем или отклоняем данную гипотезу.

Если мы принимаем гипотезу и пациент действительно болен, то такая ситуация называется True Positive (TP). Если мы отклоняем гипотезу и пациент действительно здоров, то это случай True Negative (TN). Если мы приняли гипотезу, а пациент здоров, то это False Positive (FP) (или ошибка 2-ого рода). Если отклонили гипотезу, а пациент болен, то это самый худший случай False Negative (FN) (ошибка 1-ого рода) [5].

Для оценки результатов эффективности диагностики мы использовали следующие статистические показатели:

1. Ассурасу (точность) – число правильно поставленных диагнозов к общему числу пациентов в выборке:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Precision – число правильно определенных болезней к общему числу предсказанных болезней:

$$precision = \frac{TP}{TP + FP}$$

3. Recall (чувствительность) – число правильно определенных болезней к общему числу реально больных людей:

$$recall = \frac{TP}{TP + FN}$$

4. Specificity (специфичность) – число правильно определенных здоровых людей к общему числу реально здоровых людей:

$$specificity = \frac{TN}{TN + FP}$$

5. F_1 - score (F_1 -мера) – гармоническое среднее Precision и Recall. Лучшее значение имеет в 1, худшее в 0. Характеризует информацию о точности и полноте нашего алгоритма:

$$F_1 = 2 \frac{precision * recall}{precision + recall} = \frac{2 * TP}{2 * TP + FN}$$

Дополнительно стоит отметить, что статистические показатели эффективности диагностики в задачах многоклассовой классификации имеют немного другой вид. Они делятся на micro-average, где учитывается вклад каждого класса и затем рассчитывается среднее и macro-average, где считается метрика независимо для каждого класса и берется среднее.

3.3. Результаты экспериментов

Начнем с рассмотрения ритмов сердца. Это многоклассовая задача с 4 классами: наличие синусового ритма, фибрилляций, трепетаний или другого несинусового ритма. В результате

обучения сети для данной задачи имеется 4 взаимно исключающих друг друга диагноза. Результаты в Таблице 2.

Таблица 2. Результаты работы нейронной сети для диагностики ритмов.

Метрика	Значение метрики на обучающей выборке	Значение метрики на тестовой выборке
<i>Accuracy</i>	0.9776	0.9096
<i>Precision (micro)</i>	0.9776	0.9096
<i>Recall (micro)</i>	0.9776	0.9096
$F_1 - score (micro)$	0.9776	0.9096
<i>Specificity (micro)</i>	0.9925	0.9699
<i>Precision (macro)</i>	0.9553	0.7245
<i>Recall (macro)</i>	0.9861	0.6482
$F_1 - score (macro)$	0.9702	0.6722
<i>Specificity (macro)</i>	0.9916	0.9548

Получены достаточно неплохие показатели определения ритма. Отлично определяются фибрилляции, синусовые ритмы. Трепетания дают неплохой результат, но класс очень маленький (скудная база), отсюда и спад в значениях метрик. Аналогично и другие не синусовые мало представимые ритмы, которые уменьшают итоговое значение метрик.

Если сравнить общие цифры, полученные нами и в статье, то наши числа местами превосходят значения из публикации. Однако не стоит забывать, что авторы статьи выделяли конкретные несинусовые ритмы, у нас же таких диагнозов 2: фибрилляции и трепетания (причем любые из них, а не какие-то конкретные). Это ограничение сделано в силу ограниченности нашей базы данных.

Итог: результат, полученный из сверточной сети, нас вполне удовлетворяет.

Далее следует диагностика наличия или отсутствия гипертрофии. Результаты эксперимента представлены в таблице 3.

Таблица 3. Результаты работы нейронной сети для диагностики гипертрофий.

Метрика	Значение метрики на обучающей выборке	Значение метрики на тестовой выборке
<i>Accuracy</i>	0.9529	0.741
<i>Precision</i>	0.9646	0.8205
<i>Recall</i>	0.9571	0.8136
$F_1 - score$	0.9608	0.817
<i>Specificity</i>	0.9467	0.5625

В целом получили неплохие результаты в определении самого заболевания. Однако есть проблемы с тем, что сеть периодически называет здорового пациента больным.

Следующими на очереди следуют экстрасистолы. Мы пытаемся определить наличие любой экстрасистолы у пациента. Результаты в Таблице 4.

Таблица 4. Результаты работы нейронной сети для диагностики экстрасистолы.

Метрика	Значение метрики на обучающей выборке	Значение метрики на тестовой выборке
<i>Accuracy</i>	0.9918	0.8253
<i>Precision</i>	0.8452	0.2154
<i>Recall</i>	0.9861	0.6667
$F_1 - score$	0.9103	0.3256
<i>Specificity</i>	0.992	0.836

Получили средние результаты в чувствительности и специфичности, но очень низкий precision. Это говорит о том, что здоровым пациентам ставится наличие заболевания.

Рассмотрим диагностику наличия или отсутствия АВ-блокад. Это также бинарная задача, результаты которой представлены в Таблице 5.

Таблица 5. Результаты работы нейронной сети для диагностики АВ-блокад.

Метрика	Значение метрики на обучающей выборке	Значение метрики на тестовой выборке
<i>Accuracy</i>	0.9306	0.9096
<i>Precision</i>	0.4244	0.3235
<i>Recall</i>	1	0.6111
$F_1 - score$	0.5959	0.4231
<i>Specificity</i>	0.9268	0.9268

Получились довольно средние результаты, где можно выделить высокую специфичность. Однако это сложная патология для диагностики в связи с небольшим количеством заболеваний в обучающей выборке и дальнейшее развитие нейронной сети будет связано с развитием диагностики АВ-блокад.

Далее идет крупный блок диагностики блокад ножек пучка Гиса. Каждая из задач является бинарной и определяет наличие или отсутствие заболевания. Эти классы не являются взаимно исключающими. Результаты диагностики собраны вместе и приведены в Таблице 6.

Таблица 6. Результаты работы нейронной сети для диагностики блокад ножек пучка Гиса.

Диагноз	Метрика	Значение метрики на обучающей выборке	Значение метрики на тестовой выборке
Неполная блокада правой ножки пучка Гиса	<i>Accuracy</i>	0.9265	0.6958
	<i>Precision</i>	0.6547	0.2243
	<i>Recall</i>	0.9561	0.5714
	$F_1 - score$	0.7772	0.3221
	<i>Specificity</i>	0.9219	0.7138
Неполная блокада левой ножки пучка Гиса	<i>Accuracy</i>	0.9829	0.9428
	<i>Precision</i>	0.3778	0.0667
	<i>Recall</i>	0.9444	0.1667
	$F_1 - score$	0.5397	0.0952
	<i>Specificity</i>	0.9834	0.9571
Блокада задней ветви левой ножки пучка Гиса	<i>Accuracy</i>	0.9988	100
	<i>Precision</i>	0.3333	0
	<i>Recall</i>	1	0
	$F_1 - score$	0.5	0
	<i>Specificity</i>	0.9988	0
Блокада передней ветви левой ножки пучка Гиса	<i>Accuracy</i>	0.9547	0.8404
	<i>Precision</i>	0.6765	0.3333
	<i>Recall</i>	1	0.8276
	$F_1 - score$	0.807	0.4752
	<i>Specificity</i>	0.95	0.8416
Полная блокада правой ножки пучка Гиса	<i>Accuracy</i>	0.9794	0.8795
	<i>Precision</i>	0.6429	0.2157
	<i>Recall</i>	1	1
	$F_1 - score$	0.7826	0.3548
	<i>Specificity</i>	0.9786	0.8754
Полная блокада левой ножки пучка Гиса	<i>Accuracy</i>	0.9818	0.9578
	<i>Precision</i>	0.6173	0.2778
	<i>Recall</i>	1	0.8333
	$F_1 - score$	0.7634	0.4167
	<i>Specificity</i>	0.9812	0.9601

Из результатов видно сразу, что для определения блокады задней ветви левой ножки пучка Гиса и неполной блокады левой ножки пучка Гиса не хватает данных, на которых бы сеть могла учиться и дать результат. Получены средние (~60%) результаты для неполной блокады правой ножки пучка Гиса. Неплохо определяется блокада передней ветви левой ножки пучка Гиса. Полные блокады определяются с весьма неплохими показателями качества.

Последние диагнозы, которые определяет наша нейронная сеть – электрические оси. Здесь сразу 9 классов. Это мультиклассовая задача. В Таблице 7 приведены результаты диагностики.

Таблица 7. Результаты работы нейронной сети для диагностики электрических осей сердца.

Метрика	Значение метрики на обучающей выборке	Значение метрики на тестовой выборке
<i>Accuracy</i>	0.9453	0.5994
<i>Precision (micro)</i>	0.9453	0.5994
<i>Recall (micro)</i>	0.9453	0.5994
<i>F₁ – score (micro)</i>	0.9453	0.5994
<i>Specificity (micro)</i>	0.9922	0.9428
<i>Precision (macro)</i>	0.9161	0.3933
<i>Recall (macro)</i>	0.9735	0.4586
<i>F₁ – score (macro)</i>	0.9415	0.4153
<i>Specificity (macro)</i>	0.9916	0.9353

В нашей выборке наиболее ярко выражены 4 класса из 9. Остальные имеют во много раз меньшее число представителей. В целом классы заболеваний выявляются неплохо, однако здесь есть случаи, когда электрические оси, близкие друг к другу (из ярко выраженных 4 классов) путаются нейронной сетью.

4. Заключение

В качестве заключения отметим, что нами создана сверточная нейронная сеть для диагностики 6 классов различных заболеваний. Мы можем определять ритм с показателями, превышающими 90%, гипертрофии с показателями, превышающими 81%, экстрасистолию с показателем ~67%, АВ-блокаду сердца с вероятностью 61%, электрическую ось сердца с показателем ~60% и некоторые из блокад ножек пучка Гиса с весьма неплохими показателями.

Таким образом, прогресс налицо. Часть диагнозов уже сейчас определяется с хорошей вероятностью (ритмы, гипертрофии, некоторые блокады ножек пучка Гиса). Остальные диагнозы следует определять точнее, улучшая модель и корректируя разметку исходных данных.

Если сравнивать результаты, полученные нами с результатами из работы [2], то мы получили показатели лучше, чем в публикации. Интересно здесь также то, что получены результаты от модели, архитектура которой проще и интуитивно понятнее для человека. Однако мы не можем диагностировать ритмы так детально, т.к. наша база не достаточно широка.

Дальнейшая работа будет посвящена решению задачи диагностики остальных сердечно-сосудистых заболеваний и совершенствованию того, что уже имеется (электрические оси, АВ-блокады и т.п.).

Литература

1. Зудбинов Ю.И. Азбука ЭКГ. Издание 3. Ростов-на-Дону: «Феникс», 2003. 160 с. - ISBN 5-222-02964-6.
2. Rajpurkar P., Hannun Y. A., Haghpanahi M., Bourn C., Ng Y. A. Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks, 2017. DOI: <https://doi.org/10.48550/arXiv.1707.01836>.
3. Wagner, P., Strodthoff, N., Bousseljot, R.-D., et al. PTB-XL: A Large Publicly Available ECG Dataset. Scientific Data, 2020. DOI: <https://doi.org/10.1038/s41597-020-0495-6>.

4. Moskalenko, V., Zolotykh, N., Osipov, G. Deep learning for ECG segmentation, International Conference on Neuroinformatics (Cham: Springer), 246–254. 2019. DOI: 10.1007/978-3-030-30425-6_29
5. Золотых Н. Ю. Курс лекций по Машинному Обучению и Анализу Данных, 2022. URL: http://www.uic.unn.ru/~zny/ml/Course/ml_pres.pdf (дата обращения: 05.04.2024).
6. Ng A. Machine Learning Specialization. 2012. URL: https://www.deeplearning.ai/courses/machine-learning-specialization/?utm_medium=referral&utm_source=andrew-website (дата обращения: 10.04.2024).
7. Воронцов К.В. Курс «Машинное Обучение». 2019. URL: <https://ya-r.ru/2020/05/07/vorontsov-kurs-mashinnoe-obuchenie-2019-shkola-analiza-dannyh/> (дата обращения: 01.04.2024).

Об исследовании нового класса модельных задач для проведения экспериментов с методами условной глобальной оптимизации

Е.С. Пинежанин, К.А. Баркалов

Нижегородский государственный университет им. Н.И. Лобачевского

В работе приводятся результаты вычислительных экспериментов на новом классе модельных задач, основанном на подходе, в рамках которого задача оптимизации представляется как задача подбора параметров некоторого дифференциального уравнения. Такой подход позволяет независимо от числа переменных наглядно наблюдать процесс работы алгоритма оптимизации.

Ключевые слова: глобальная оптимизация, дифференциальное уравнение, тестовые задачи.

1. Введение

Решение многих задач, возникающих в различных сферах человеческой деятельности, связано с поиском лучших значений некоторых параметров, например, массы, плотности вещества, геометрического размера некоторого объекта, соотношения химических веществ в смеси и т. д., причем часто требуется найти не просто локально-оптимальное решение задачи, а лучшее из всех – глобальное.

Одним из типов объектов, который может исследоваться в ходе решения задачи оптимизации, является «*черный ящик*». Для данного объекта характерна такая черта, как неизвестность или сложность внутреннего представления.

В реальных задачах это выражается в отсутствии аналитической формулы для целевой функции, т.е. взятие производных и вычисление градиента становятся невозможными. Нам доступен только «вход», на который мы можем подать точку (вектор варьируемых параметров), и «выход», на котором можно получить значение целевой функции в точке, которую подали на «вход».

Требования, описанные выше, привели к возникновению класса численных методов глобальной оптимизации.

Одним из подходов к сравнению эффективности таких алгоритмов является применение этих методов для решения множества тестовых задач из некоторого сконструированного класса. Это позволяет оценивать операционные характеристики методов (вероятности нахождения глобального экстремума за заданное число шагов поиска) и тем самым характеризовать эффективность алгоритма.

Как правило генераторы тестовых задач имеют один общий недостаток, они порождают непосредственно саму оптимизируемую функцию, т.е. для задач с размерностью $N > 2$ не удастся наглядно наблюдать сам процесс оптимизации. В связи с этим представляет интерес новый подход, инициированный в работе [1]. В рамках предлагаемого подхода оптимизационная задача возникает как задача идентификации (подбора) параметров дифференциального уравнения, решение которого $u(t)$ является одномерной кривой, что позволяет независимо от числа переменных наглядно наблюдать как текущее, так и окончательное решение задачи.

2. Постановка задачи

В работе используется параллельный алгоритм для поиска глобального минимума многоэкстремальной функции $\varphi(y)$ при наличии m ограничений типа неравенств с невыпуклыми левыми частями $g_j(y)$, $1 \leq j \leq m$. Предполагается, что функции, входящие в постановку задачи, удовлетворяют условию Липшица с априори неизвестными константами L_j .

Таким образом, задача рассматривается в виде

$$\begin{aligned}\varphi(y^*) &= \min\{\varphi(y), y \in D, g_j(y) \leq 0, 1 \leq j \leq m\}, \\ D &= \{y \in R^N: a_i \leq y_i \leq b_i, i = \overline{1, N}\}.\end{aligned}$$

Предположение липшицевости функций задачи является типичным для многих подходов к разработке алгоритмов глобальной оптимизации [2]. В отличие от широко распространенных метаэвристических алгоритмов, методы липшицевой глобальной оптимизации обладают детерминированным поведением и более быстрой сходимостью к решению задачи [3].

Используемый в работе подход к поиску оптимума основан на редукции исходной многомерной задачи к эквивалентной ей одномерной задаче с использованием кривых, заполняющих пространство (кривых Пеано) $y(x)$. Непрерывное однозначное отображение $y(x)$, переводящее отрезок $[0,1]$ в N -мерный гиперкуб D , позволяет свести многомерную задачу условной минимизации в области D к одномерной задаче условной минимизации на отрезке $[0,1]$

$$\varphi(y(x^*)) = \min\{\varphi(y(x)): x \in [0,1], g_j(y(x)) \leq 0, 1 \leq j \leq m\}$$

Решение редуцированной задачи одномерной задачи может быть выполнено эффективными алгоритмами минимизации функции одной переменной. В качестве такого алгоритма, основанного на оригинальной схеме учета ограничений и не использующего идей метода штрафных функций, исследовался индексный метод глобальной оптимизации [3].

Распараллеливание метода было организовано следующим образом. Каждой подобласти поиска ставится в соответствие число, называемое *характеристикой*. Она определяет перспективность данной подобласти для проведения в ней нового испытания. На каждой итерации алгоритма определяется p лучших подобластей (в соответствии с их характеристиками), в каждой из которых параллельно проводится одно поисковое испытание. Указанный способ обладает общностью (т.к. может быть применен для широкого класса divide-the-best алгоритмов) и эффективностью (т.к. проведение испытаний в прикладных задачах является трудоемкой операцией).

3. Описание класса задач

В качестве дифференциального уравнения рассмотрим математическую модель движения заряженной частицы в магнитном поле:

$$m\ddot{u} = -eu + F,$$

где m – масса частицы, $u(\tau)$ – текущее положение частицы в момент времени $\tau \geq 0$, $-eu$ – величина силы, притягивающей частицу к началу координат, F – величина внешней силы, направленной вдоль оси u .

Общим решением этого уравнения будет являться одномерная кривая

$$u(t, \omega, c) = \sum_{i=0}^n [c_{2i+1} \sin(\omega_i t) + c_{2i+2} \cos(\omega_i t)].$$

В задаче требуется, чтобы кривая решения $u(t, \omega, c)$ проходила бы через выделенные области (см. рис. 1), причем в последней из них имела бы максимальный наклон, т.е. решается задача

$$|\dot{u}(t, \omega, c)| \rightarrow \max$$

Дополнительно учтем симметричность кривой $u(t, \omega, c)$, что позволит упорядочить по возрастанию компоненты вектора частот ω , чтобы исключить дублирующиеся решения, соответствующие вектору частот ω с одинаковыми компонентами в разном порядке. Более того, естественным является предположение, что частоты в управляющем воздействии должны быть не просто упорядочены, а различаться на некоторую положительную величину, т.к. реальное физическое устройство способно обеспечить генерацию управляющего сигнала лишь с некоторой точностью. Данное предположение может быть представлено в виде требования о выполнении неравенства

$$\omega_{i-1}(1 + \alpha) - \omega_i(1 - \alpha) \leq 0, 1 \leq i \leq n,$$

где $0 < \alpha < 1$ является параметром, отражающим точность генерации сигнала управляющим устройством.

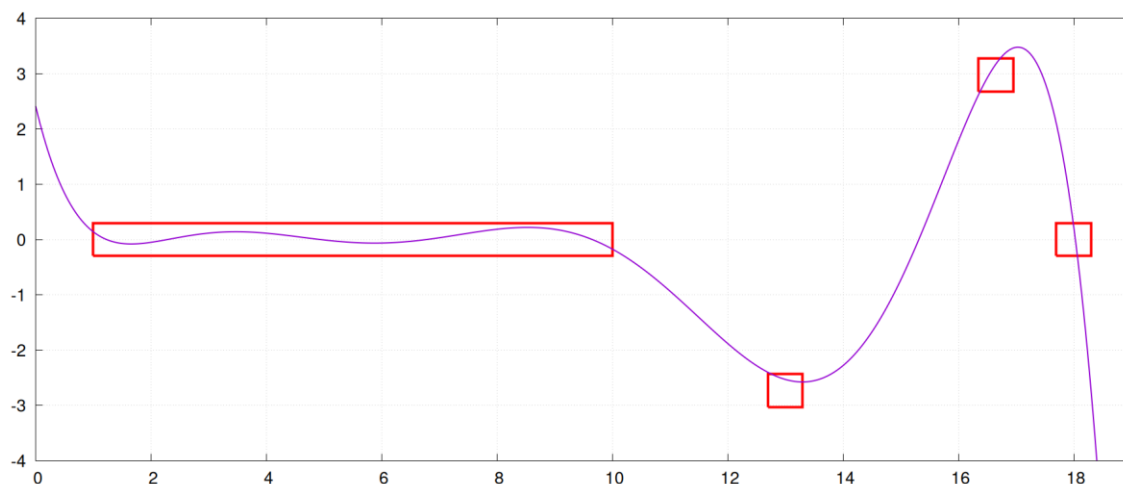


Рис. 1. Пример решения задачи подбора параметров дифференциального уравнения.

3. Результаты вычислительных экспериментов

Первоначальной задачей при исследовании нового класса задач была проверка возможности наглядного наблюдения процесса поиска оптимального решения. Для этого были вычислены оптимальные точки при некоторых фиксированных значениях количества испытания для одной задачи из семейства. После были построены одномерные кривые $u(t, \omega, c)$ для каждого случая. Полученные кривые представлены на рис. 2.

На графике отчетливо видно, как с увеличением числа испытаний метода кривая увеличивает угол наклона в последнем “окне”, значит метод находит с каждым разом находит более точное решение и приближается к глобальному оптимуму, что соответствует корректному ходу работы алгоритма оптимизации.

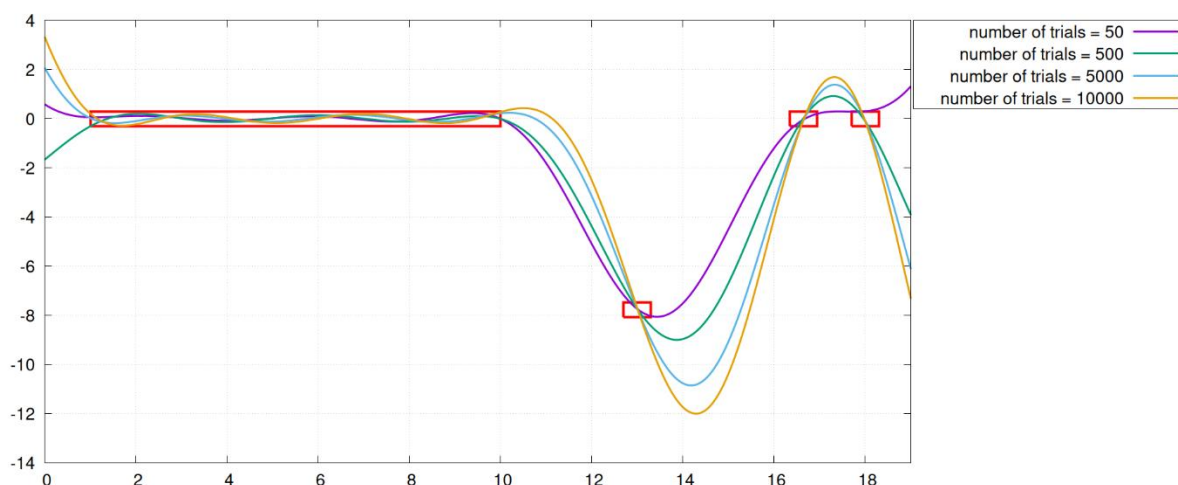


Рис. 2. Решения задачи, полученные после проведения K испытаний, $K = 50, 500, 5000, 10000$.

Дальнейшим шагом к исследованию новых модельных задач была генерация выборки из нескольких задач и вычисление операционной характеристики метода. Для проведения экспериментов была получена выборка из 20 задач. Модельные задачи решались с помощью

параллельного алгоритма, дополнительно оценивалось ускорение по времени для запусков с различным количеством вычисляемых точек на одной итерации.

Программная реализация параллельного алгоритма глобальной оптимизации выполнена на C++ с использованием OpenMP. Эксперименты проводились на узле суперкомпьютера «Лобачевский» (2 процессора AMD EPYC 7742 2.25 GHz, 64 физических ядра, до 128 параллельных потоков в режиме MultiThreading).

Таблица 1. Количество точек на итерации p , среднее число итераций K_{av} , время работы T_{av} и S_{av} ускорение параллельного алгоритма

p	K_{av}	T_{av} , сек.	S_{av}
1	100 001	23.3	----
16	6 251	12.32	1.9
32	3 126	10.28	2.3
64	1 563	6.5	3.6
128	781	3.8	6.1

Результаты, приведенные в таблице, показывают уменьшение среднего числа итераций, необходимого для решения одной задачи, за счет параллельного проведения нескольких испытаний в разных процессах, однако ускорение по времени достигается небольшое.

Во время детального рассмотрения причин такого малого ускорения метода было обнаружено «узкое» место исследуемого класса: около 70% всех испытаний заключаются только в проверке, что компоненты вектора ω упорядочены и различаются на некоторую положительную величину и не производят вычисления более «тяжеловесных» ограничений, а на вычисление целевой функции приходится менее 1% от общего числа испытаний. Это представляет серьезную проблему при использовании параллельного алгоритма, так как само распределение вычислительной нагрузки по потокам добавляет накладные расходы, а, следовательно, вычисления должны превосходить их по времени, чтобы мы смогли получить уменьшение времени работы алгоритма. Дополнительным фактором низкой эффективности представленного алгоритма является синхронная схема распараллеливания, она не позволяет переходить методу к следующей итерации, пока не будут подсчитаны все точки на текущей, что при различной вычислительной нагрузке на каждый поток дает уменьшение эффективности распараллеливания.

В качестве решения этой проблемы было предложено убрать требование к компонентам вектора ω быть упорядоченными, так как исходная целевая функция симметричная и любая перестановка координат вектора ω будет давать одинаковое значение целевой функции.

Таблица 2. Количество точек на итерации p , среднее число итераций K_{av} , время работы T_{av} и ускорение S_{av} параллельного алгоритма

p	K_{av}	T_{av} , сек.	S_{av}
1	64 617	49.8	----
16	3 971	20.8	2.4
32	2 326	16.5	3.0
64	1 247	11.0	4.5
128	561	5.9	8.5

Модификация тестовой модельной задачи позволила уменьшить число испытаний с проверкой только простых ограничений до 40%, к тому же позволила улучшить сходимость алгоритма к оптимуму, что можно увидеть из уменьшения среднего числа итераций, однако ускорение по времени все еще достигается небольшое, что дает возможность для дальнейшего исследования и улучшения модельного класса задач.

4. Заключение

В ходе исследования модельной задачи были проведены вычислительные эксперименты, показывающие возможность наглядно наблюдать ход решения задачи. Произведены вычисления операционной характеристики параллельного метода. Найдено «узкое» место нового класса задач – различная вычислительная сложность ограничений, кроме того, показан один из возможных путей решения такой проблемы.

Предложенная в работе модификация класса не обеспечила достаточное ускорение. Синхронная схема распараллеливания вычислений также не позволила в полной мере использовать вычислительные ресурсы, поэтому направлением дальнейших исследований будет использование асинхронной схемы распараллеливания вычислений. Асинхронная схема позволяет равномерно распределять нагрузку на потоки для задач, в которых время проведения испытаний в разных точках области поиска является различным.

Литература

1. Barkalov K., Strongin R. Test problems for parallel algorithms of constrained global optimization // Lecture notes in computer science. 2017. vol. 10556. P. 18–33.
2. Евтушенко Ю.Г., Посыпкин М.А. Метод неравномерных покрытий для решения задач многокритериальной оптимизации с заданной точностью // Автомат. и телемех. 2014. № 6, С. 49–68.
3. Sergeyev Y., Kvasov D., Mukhametzhano M. On the Efficiency of Nature-Inspired Metaheuristics in Expensive Global Optimization with Limited Budget // Scientific Reports. vol. 8(1). Art. No. 435.
4. Баркалов К.А. Оценки эффективности параллельного индексного метода глобальной оптимизации // Вестник Нижегородского университета им. Н.И. Лобачевского. 2011. №3(2), С. 13-19.

Determining the presence of heart disease by ECG in the PTB-XL dataset using machine learning

V.V. Razin, A.A. Krasnov

Lobachevsky State University of Nizhny Novgorod

Determining whether a person has heart problems by ECG recording is an extremely important and relevant task in medicine. The paper explores the use of machine learning to solve this problem. It can be seen that machine learning methods show good indicators of the predictive ability of the presence of heart disease by analyzing the ECG record. The use of ensembles makes it possible to improve the predictive ability of the program, and 5 types of ensembles demonstrate the greatest accuracy at once. The addition of artificial models also improves the accuracy of the ensembles. Using both a deep and machine learning model as participants allows you to bypass the accuracy of the one that is built only on machine learning models. However, adding artificial models to these ensembles equalizes the accuracy of their predictions. The obtained accuracy of these ensembles is higher than the accuracy of ensembles on deep learning models, which justifies the use of machine learning in this problem. This software package can be implemented into various medical diagnostic systems is possible, for example, the real-time system "Cardiomayak", developed at the National Research Lobachevsky State University of Nizhny Novgorod.

Keywords: machine learning, ECG diagnostics, ensemble, artificial intelligence, binary classification, neural network features, deep learning, recurrent neural network, convolutional neural network.

1. Introduction

Determining whether a person has heart problems by ECG recording is an extremely important and relevant task in medicine. The paper explores the use of machine learning to solve this problem.

PTB-XL is one of the world's largest open heart disease datasets hosted by PhysioNet. The problem of binary classification of the presence of heart diseases or their absence according to the ECG records from this dataset is solved.

Approaches for the classification of time series are described in many works [1-4] using various architectures of neural networks and methods [5-10].

2. Data and methods

2.1 Data

Machine learning methods worked with the features of neural networks that were used in solving the same problem, but with deep learning approaches.

2.2 Binary classification metrics

As a metric for assessing the predictive ability of models, the accuracy is used, which shows the proportion of correctly predicted predictions of trained machine methods and ensembles.

2.3 Methods

In previous work, the following neural networks were used:

- conventional convolutional network (hereinafter CNN);
- separable convolutional network (hereinafter SCNN);
- AlexNet-based network (hereinafter AlexNet);

- combination of a convolutional network with bidirectional LSTM and GRU (hereinafter CBB);
- Inception-based network (hereinafter Inception);
- recurrent network LSTM (hereinafter LSTM);
- GRU recurrent network (hereinafter GRU);
- combination of LSTM and GRU (hereinafter LG);
- VGG16 based network (hereinafter VGG16);
- Xception-based network (hereinafter Xception).

In this work, the following machine methods were used:

- logistic regression (hereinafter LR);
- logistic regression with cross-validation (hereinafter LRC);
- perceptron (hereinafter PRC);
- LightGBM (hereinafter LGB);
- XGBoost (hereinafter XGB);
- CatBoost (hereinafter CB);
- stochastic gradient descent (hereinafter SGD);
- k-nearest neighbor method (hereinafter KNN);
- passive-aggressive classifier (hereinafter PAC);
- linear discriminant analysis (hereinafter LDA);
- decision tree (hereinafter DT)
- extremely randomized trees (hereinafter ETS);
- random forest (hereinafter RF);
- Ridge regressor (hereinafter RC).

Also in this work, 7 types of ensembles were used:

- arithmetic mean of prediction probabilities of 2 classes (hereinafter mla)
- weighted average (the greater the accuracy, the greater the contribution) of prediction probabilities of 2 classes (hereinafter mlk)
- arithmetic mean of rounded prediction probabilities of 2 classes (hereinafter fmla)
- weighted average (the greater the accuracy, the greater the contribution) of rounded prediction probabilities of 2 classes (hereinafter fmlk)
- arithmetic mean of class number predictions (hereinafter nca)
- weighted average (the greater the accuracy, the greater the contribution) of class number predictions (nck)
- mode (if parity, then the first element is returned) among the class number predictions (hereinafter ncd)

3. Experimental results

3.1 Description of the task

2 classes are predicted (HEALTH and SICK). The HEALTH class is all entries from PTBXL that only have the NORM class have a 1 and all others have zeros. The SICK class is assigned to the remaining entries.

3.2 The ensemble with the best predictive power

As a result of the experiments, an ensemble was found consisting of 2 machine methods: CB, KNN; 3 deep learning models: Inception, LSTM and CNN; and 1 artificial model, that always return 1 as number class in prediction. The ABS of this ensemble is 89.90% (rounded to the nearest hundredth).

3.3 Experimental details

The determination of the most suitable models for classification is carried out experimentally. After changing the hyperparameters and subsequent training, the obtained accuracy is compared with the previous results. If the accuracy has increased, then the value of the changed hyperparameters is fixed. Also, the selected metric is used to compare the machine learning models themselves, as well as various ensembles.

At the beginning, the KNN method is trained on all possible combinations of features of all neural networks. The best combination is considered to be the one on which this model will show the greatest accuracy. After a complete enumeration, such a combination became signs from the following neural networks: AlexNet, VGG16, Inception, LSTM, GRU, SCNN and LG.

Further, on this set of features, all machine learning methods were trained and optimal hyperparameters were selected for them. Accuracy of models after selection of optimal hyperparameters are shown in Table 1 (highest accuracy for each case in bold).

Table 1. Accuracy for machine methods (as a percentage, rounded to hundredths)

Model name	Accuracy
LR	88.48
LRC	88.48
PRC	88.71
LGB	88.85
XGB	88.57
CB	88.71
DT	88.48
ETS	89.17
SGD	89.12
KNN	88.89
PAC	88.34
LDA	88.39
RF	89.54
RC	88.25

You can see that the RF machine method shows the highest accuracy.

For all ensemble types, such a set of participants is searched (by exhaustive enumeration), which in total produces the largest accuracy. Table 2 shows the accuracy scores for each ensemble. The highest accuracy score for each case is shown in bold.

Table 2. Accuracy of ensembles from all machine methods

Ensemble name	Accuracy
m1a	89.54
m1k	89.54
fmla	89.72
fmlk	89.72
nca	89.72
nck	89.72
ncd	89.72

It can be seen that the ensembles fmla, fmlk, nca, nck, ncd show the highest accuracy.

Next, to the models that were the best in the ensemble, we add 2 artificial models that always return the same class number in their predictions (each one is different). Adding an artificial model to the members of the best ensemble, which always returns a class number of 1, increases the accuracy of the ensemble to 89.90%.

After that, the ncd-type ensemble is built from the models that were part of the best deep learning and machine learning ensembles in terms of accuracy. Its accuracy is 89.86%.

Finally, to the models that were the best in the ensemble, we add 2 artificial models that always return the same class number in their predictions (each one is different). Adding an artificial model to the members of the best ensemble, which always returns a class number of 1, increases the accuracy of the ensemble to 89.90%.

4. Conclusions

It can be seen that machine learning methods show good indicators of the predictive ability of the presence of heart disease by analyzing the ECG record. The use of ensembles makes it possible to improve the predictive ability of the program, and 5 types of ensembles demonstrate the greatest accuracy at once. The addition of artificial models also improves the accuracy of the ensembles. Using both a deep and machine learning model as participants allows you to bypass the accuracy of the one that is built only on machine learning models. However, adding artificial models to these ensembles equalizes the accuracy of their predictions. The obtained accuracy of these ensembles is higher than the accuracy of ensembles on deep learning models, which justifies the use of machine learning in this problem.

This software package can be implemented into various medical diagnostic systems is possible, for example, the real-time system "Cardiomayak", developed at the National Research Lobachevsky State University of Nizhny Novgorod.

In the future, it is planned to consider other, more complex architectures of neural networks, the use of new unproven methods of machine learning, as well as take other types of ensembles. It is also planned to consider more complex diagnostic tasks and introduce explainable artificial intelligence to solve the tasks.

Acknowledgements

Results are supported by Ministry of Science and Education of Russian Federation, project FSWR-2024-0005.

References

1. Razin V. [et al.]. Solving the Problem of Diagnosing a Disease by ECG on the PTB-XL Dataset Using Deep Learning // International Conference on Neuroinformatics. – Cham: Springer Nature Switzerland, 2023. С. 13-21.
2. Разин В.В., Краснов А.А., Золотых Н.Ю. Глубокое обучение в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-ХЛ // Математика и ИТ – вместе в цифровое будущее. Сборник трудов Молодежной школы (Н. Новгород, 25–29 апреля 2022 г.). Нижний Новгород: Изд-во Нижегородского госуниверситета, 2022. С.78-86.
3. Краснов А.А., Разин В.В., Золотых Н.Ю. Машинное обучение в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-ХЛ // Математика и ИТ – вместе в цифровое будущее. Сборник трудов Молодежной школы (Н. Новгород, 25–29 апреля 2022 г.). Нижний Новгород: Изд-во Нижегородского госуниверситета, 2022. С. 44-50.
4. Разин В.В., Краснов А.А. Использование искусственного интеллекта в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-ХЛ // Путь в науку: прикладная математика, информатика и информационные технологии. Тезисы докладов Всероссийской молодёжной научно-практической конференции (Ярославль, 17–21 апреля 2023 г.). Ярославль: Ярославский государственный университет им. П.Г. Демидова, 2023. С. 93-96.
5. Разин В.В., Краснов А.А. Применение машинного обучения в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-ХЛ // Математика и ИТ – вместе в цифровое будущее. Сборник трудов II Молодежной школы (Нижний Новгород, 24–28 апреля 2023 года). Нижний Новгород: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского, 2023. С. 73-79.
6. Разин В.В., Краснов А.А. Применение глубокого обучения в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-ХЛ // Математика и ИТ – вместе в цифровое будущее. Сборник трудов II Молодежной школы (Нижний Новгород, 24–28 апреля 2023 года). Нижний Новгород: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского, 2023. С. 66-72.
7. Разин В. В., Краснов А. А. Использование глубокого обучения в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-ХЛ // Биосистемы: организация, поведение, управление : Тезисы докладов 76-й Всероссийской с международным участием школы-конференции молодых ученых, Нижний Новгород, 11–14 апреля 2023 года. – Нижний Новгород: Нижегородский госуниверситет им. Н.И. Лобачевского, 2023. С. 267.
8. Разин В. В., Краснов А. А., Карчков Д. А. Классификация болезней сердца на наборе данных РТВ-ХЛ при помощи машинного обучения // Академический форум молодых ученых стран Большой Евразии «Континент науки»: Сборник тезисов докладов, Москва, 01–04 ноября 2023 года. – Москва: Центр научно-технических решений, 2023. С. 570-573.
9. Краснов А. А., Разин В. В., Карчков Д. А. Классификация болезней сердца на наборе данных РТВ-ХЛ при помощи глубокого обучения // Академический форум молодых ученых стран Большой Евразии «Континент науки»: Сборник тезисов докладов, Москва, 01–04 ноября 2023 года. – Москва: Центр научно-технических решений, 2023. С. 548-551.
10. Razin V. [et al.]. Determining the presence of heart disease by ecg in the PTB-XL dataset using deep learning // Математическое моделирование и суперкомпьютерные технологии: Труды XXIII Международной конференции, Нижний Новгород, 13–16 ноября 2023 года. – Нижний Новгород: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского, 2023. С. 173-176.

Solving the problem of diagnosing a disease by ECG on the PTB-XL dataset using machine learning

V.V. Razin, A.A. Krasnov

Lobachevsky State University of Nizhny Novgorod

Diagnosis by electrocardiogram is an extremely urgent and important task, the quality, timeliness and speed of the solution of which depend on the life and health of people. Prediction of major disease classes can be useful in identifying common heart problems in patients. The article explores the use of machine learning as a tool for solving the problem of determining pathological ECG signals with markers of myocardial infarction, hypertrophy, conduction disturbances, and changes in ST segment morphology. In the course of the experiments, the positive effect of using the features of neural networks for solving the same problem was established. Finding thresholds for replacements of trained methods also improves predictive power. Compiling different ensembles on trained models also allows you to increase your predictive ability. The addition of artificial models also improves the classifying ability of ensembles. Returning a random number in the absence of a single mode also improves the accuracy of the ensemble. The use of both machine learning and deep learning models also has a positive effect on the predictive ability of ensembles.

Keywords: machine learning, ECG diagnostics, ensemble, artificial intelligence, multilabel classification, neural network features, deep learning, recurrent neural network, convolutional neural network.

1. Introduction

Diagnosis by electrocardiogram is an extremely urgent and important task, the quality, timeliness and speed of the solution of which depend on the life and health of people. Prediction of major disease classes can be useful in identifying common heart problems in patients. The article explores the use of machine learning as a tool for solving the problem of determining pathological ECG signals with markers of myocardial infarction, hypertrophy, conduction disturbances, and ST segment morphology changes.

The procedure for diagnosing an electrocardiogram using software is considered. The PTB-XL dataset is one of the largest open heart disease datasets in the world hosted by PhysioNet. On this data set, the problem of diagnosing an electrocardiogram of classifying the main subclasses of diagnoses is solved. Machine learning is used as an ECG analysis tool that demonstrates good performance in solving classification problems in their various varieties, as well as in ECG diagnostic problems.

Approaches for the classification of time series are described in many works [1-4] using various architectures of neural networks and methods [5-10].

2. Data and methods

2.1 Data

Machine learning methods worked with the features of neural networks that were used in solving the same problem, but with deep learning approaches.

2.2 Multilabel classification metrics

As metrics for evaluating classification models, preference was given to the ABS metric (percentage of absolutely accurately predicted diagnoses), which clearly shows the percentage of accurate predictions of a particular model. This metric shows the absolute accuracy of predictions and well distributes various machine learning methods according to their ability to predict the desired set of subclasses.

2.3 Methods

The following neural networks were used in the work:

- conventional convolutional network (hereinafter CNN);
- separable convolutional network (hereinafter SCNN);
- AlexNet-based network (hereinafter AlexNet);
- combination of a convolutional network with bidirectional LSTM and GRU (hereinafter CBB);
- Inception-based network (hereinafter Inception);
- recurrent network LSTM (hereinafter LSTM);
- GRU recurrent network (hereinafter GRU);
- combination of LSTM and GRU (hereinafter LG);
- VGG16 based network (hereinafter VGG16);
- Xception-based network (hereinafter Xception).

The networks were trained using two methods, multilabel and multiclass. A multilabel is a prediction of a vector of length 5, where each class can be present or absent, and a multiclass is a prediction of a specific set of classes, i.e. only one specific set of classes is predicted.

Top 10 predictive methods:

- stochastic gradient descent (hereinafter SGD);
- k-nearest neighbor method (hereinafter KNN);
- passive-aggressive classifier (hereinafter PAC);
- linear discriminant analysis (hereinafter LDA);
- decision tree (hereinafter DT)
- extremely randomized tree (hereinafter ET);
- extremely randomized trees (hereinafter ETS);
- random forest (hereinafter RF);
- Ridge regressor (hereinafter RC);
- RidgeCV regressor (hereinafter RCV).

Also in this work 10 types of ensembles were used:

- arithmetic mean of prediction probabilities of 5 classes (hereinafter mla);
- weighted average (the greater the accuracy, the greater the contribution) of prediction probabilities of 5 classes (hereinafter mlk);
- arithmetic mean of rounded prediction probabilities of 5 classes (hereinafter fmla);
- weighted average (the greater the accuracy, the greater the contribution) of rounded prediction probabilities of 5 classes (hereinafter fmlk);
- arithmetic mean of prediction probabilities of a set of classes (hereinafter mca);
- weighted average (the greater the accuracy, the greater the contribution) of prediction probabilities of a set of classes (hereinafter mck);
- arithmetic mean of rounded prediction probabilities of a set of classes (hereinafter fmca);
- weighted average (the greater the accuracy, the greater the contribution) of rounded prediction probabilities of a set of classes (hereinafter fmck);
- mode (if parity, then the first element is returned) among the class set number predictions (hereinafter fmcd);
- mode (if parity, then the random element is returned) among the class set number predictions (hereinafter fmcdR).

3. Experimental results

3.1 Description of the task

The problem of multilabel classification of prediction of the presence of classes (NORM, MI, HYP, STTC, CD) is being solved based on the features of neural networks that solve the same classification problem.

This task is relevant, since at the moment the selected dataset is the largest open dataset in the world, which periodically receives updates. This data size makes it possible to predict the growth of predictive ability by increasing the training sample.

3.2 The ensemble with the best predictive power

As a result of the experiments, an ensemble was found consisting of 1 machine learning methods: KNN, 2 multiclass models: CBB and LG, 2 multilabel models: LSTM and CNN and 2 artificial models that always return the following numbers of diagnoses sets: 6, 18. The ABS of this ensemble is 73.24% (rounded to the nearest hundredth).

3.3 Experimental details

An experimental method was chosen as a research method, the essence of which is to find the optimal hyperparameters of machine learning methods and check the level of their classification after the learning process. Based on the training results of the selected machine learning methods, all kinds of ensembles of various types are built and the best one is identified in terms of its classifying ability. Based on the results of the selection, the features of the neural networks Alexnet, VGG16 trained by the multilabel method and VGG16, CBB, LG trained by the multiclass method were taken as input data. The selection was made by a complete enumeration of all possible combinations of features of multiclass networks and all possible combinations of features of multiple label networks. The set that produced the maximum ABS on KNN is recognized as the best. Next, all the features from both of these best sets are taken and the best set of them is searched for by exhaustive search, which gives the highest ABS on CNN.

Next, 10 machine learning models are trained on the selected set of features of neural networks and optimal hyperparameters are selected for them by random search, exhaustive search and dichotomy methods. ABS scores of these methods are shown in Table 1 (highest ABS for each case in bold).

Table 1. ABS for machine learning methods (as a percentage, rounded to hundredths)

Method	ABS
DT	60.66
ETS	71.35
ET	62.00
SGD	68.77
KNN	70.52
PAC	67.99
LDA	67.34
RF	70.75
RC	68.03
RCC	68.17

After that, all kinds of ensembles of different types are compiled, which include either all methods, or only a part of them, which can give out the probability of belonging to classes (all methods except for PAC, RC, RCC). Table 2 shows the ABS scores for each ensemble. The highest ABS score for each case is shown in bold.

Table 2. ABS for all ensembles (as a percentage, rounded to the nearest hundredth)

Ensemble name	ABS
m1a	71.35
m1k	71.35
fmla	71.35
fmlk	71.35
mca	71.35
mck	71.35
fmca	71.35
fmck	71.35
fmcd	71.49

It can be seen that the ensemble fmcd shows the highest ABS score. It consists of ET, ETS and KNN methods.

After training the models, we can try to improve their classifying ability by finding a replacement for their prediction at a certain threshold (this rule applies to all predictions). In the worst case, the predictions will remain unchanged. Table 3 shows the ABS performance for each network and for each approach. The highest ABS score for each case is shown in bold.

Table 3. ABS for machine learning methods after improvement (as a percentage, rounded to hundredths)

Method	ABS
DT	60.66
ETS	71.35
ET	62.00
SGD	69.78
KNN	71.26
PAC	67.99
LDA	68.17
RF	70.75
RC	68.03
RCC	68.17

You can see that some models (SGD, KNN, LDA) have increased ABS after selecting replacement thresholds and the replacements themselves.

Further, all possible ensembles are built from the methods after their improvement. The fmcd ensemble shows the highest ABS score, which consists of ET, ETS, SGD, KNN, PAC. The ABS of this ensemble is 71.90% (rounded to the nearest hundredth).

The fmcd_r ensemble (with the same members) already gives out 72.18% (rounded to hundredths).

Next, we alternately add one artificial model (a model that always returns one particular set of classes as a prediction) to this set of participants and see if there is an improvement. Artificial models numbered 2, 10 and 30 improve the predictive power of the fmcd ensemble. Then we add all these artificial models to the best set of conventional machine learning models and build all kinds of fmcd type ensembles. We get a new best ensemble (according to the ABS metric), which consists of ET, ETS, KNN, PAC and artificial models with numbers 2, 10 and 30. The ABS of this ensemble is 72.09% (rounded to the nearest hundredth). Next, we look for the best ensemble (according to the ABS metric) fmcd_r for this set of participants. The ABS of this ensemble is 72.27% (rounded to the nearest hundredth).

Finally, all possible fmcd-type ensembles are compiled from models and methods that were included in the best ensembles (in terms of ABS) and to which the improvement was applied. It consists of the following members: KNN, multiclass CBB and LG and multilabel LSTM and CNN. The ABS of this ensemble is 72.92% (rounded to the nearest hundredth). Next, we alternately add one artificial model to this set of participants and see if there is an improvement. Artificial models numbered 6, 10, 18 and 30 improve the predictive power of the fmcd ensemble. Then we add all these artificial models to the best set of conventional machine learning models and build all kinds of fmcd type ensembles. We get a new best ensemble (according to the ABS metric). It consists of the following members: KNN, multiclass CBB and LG, multilabel LSTM and CNN and artificial models with numbers 6 and 18. The ABS of this ensemble is 73.15% (rounded to the nearest hundredth). Next, we look for the best ensemble (according to the ABS metric) fmcd_r for this set of participants. The ABS of this ensemble is 73.24% (rounded to the nearest hundredth).

4. Conclusions

In the course of the experiments, the positive effect of using the features of neural networks for solving the same problem was established. Finding thresholds for replacements of trained methods also improves predictive power. Compiling different ensembles on trained models also allows you to increase your predictive ability. The addition of artificial models also improves the classifying ability of ensembles. Returning a random number in the absence of a single mode also improves the accuracy of the ensemble. The use of both machine learning and deep learning models also has a positive effect on the predictive ability of ensembles.

In the future, it is planned to consider other, more complex architectures of neural networks, the use of new unproven methods of machine learning, as well as take other types of ensembles. It is also planned to consider more complex diagnostic tasks and introduce explainable artificial intelligence to solve the tasks.

Acknowledgements

Results are supported by Ministry of Science and Education of Russian Federation, project FSWR-2024-0005.

References

1. Razin V. [et al.]. Solving the Problem of Diagnosing a Disease by ECG on the PTB-XL Dataset Using Deep Learning // International Conference on Neuroinformatics. – Cham: Springer Nature Switzerland, 2023. С. 13-21.
2. Разин В.В., Краснов А.А., Золотых Н.Ю. Глубокое обучение в задачах диагностики заболевания по ЭКГ на наборе данных PTB-XL // Математика и ИТ – вместе в цифровое

будущее. Сборник трудов Молодежной школы (Н. Новгород, 25–29 апреля 2022 г.). Нижний Новгород: Изд-во Нижегородского госуниверситета, 2022. С.78-86.

3. Краснов А.А., Разин В.В., Золотых Н.Ю. Машинное обучение в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-XL // Математика и ИТ – вместе в цифровое будущее. Сборник трудов Молодежной школы (Н. Новгород, 25–29 апреля 2022 г.). Нижний Новгород: Изд-во Нижегородского госуниверситета, 2022. С. 44-50.
4. Разин В.В., Краснов А.А. Использование искусственного интеллекта в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-XL // Путь в науку: прикладная математика, информатика и информационные технологии. Тезисы докладов Всероссийской молодёжной научно-практической конференции (Ярославль, 17–21 апреля 2023 г.). Ярославль: Ярославский государственный университет им. П.Г. Демидова, 2023. С. 93-96.
5. Разин В.В., Краснов А.А. Применение машинного обучения в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-XL // Математика и ИТ – вместе в цифровое будущее. Сборник трудов II Молодежной школы (Нижний Новгород, 24–28 апреля 2023 года). Нижний Новгород: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского, 2023. С. 73-79.
6. Разин В.В., Краснов А.А. Применение глубокого обучения в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-XL // Математика и ИТ – вместе в цифровое будущее. Сборник трудов II Молодежной школы (Нижний Новгород, 24–28 апреля 2023 года). Нижний Новгород: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского, 2023. С. 66-72.
7. Разин В. В., Краснов А. А. Использование глубокого обучения в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-XL // Биосистемы: организация, поведение, управление : Тезисы докладов 76-й Всероссийской с международным участием школы-конференции молодых ученых, Нижний Новгород, 11–14 апреля 2023 года. – Нижний Новгород: Нижегородский госуниверситет им. Н.И. Лобачевского, 2023. С. 267.
8. Разин В. В., Краснов А. А., Карчков Д. А. Классификация болезней сердца на наборе данных РТВ-XL при помощи машинного обучения // Академический форум молодых ученых стран Большой Евразии «Континент науки»: Сборник тезисов докладов, Москва, 01–04 ноября 2023 года. – Москва: Центр научно-технических решений, 2023. С. 570-573.
9. Краснов А. А., Разин В. В., Карчков Д. А. Классификация болезней сердца на наборе данных РТВ-XL при помощи глубокого обучения // Академический форум молодых ученых стран Большой Евразии «Континент науки»: Сборник тезисов докладов, Москва, 01–04 ноября 2023 года. – Москва: Центр научно-технических решений, 2023. С. 548-551.
10. Razin V. [et al.]. Determining the presence of heart disease by ecg in the PTB-XL dataset using deep learning // Математическое моделирование и суперкомпьютерные технологии: Труды XXIII Международной конференции, Нижний Новгород, 13–16 ноября 2023 года. – Нижний Новгород: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского, 2023. С. 173-176.

Разработка функции для трекинга эмоций в мобильном приложении для подростков с психологическими проблемами

А.Р. Резникова

Тольяттинский государственный университет

В статье рассматривается возможность интеграции функций трекинга эмоционального состояния за день в разрабатываемое мобильное приложение для подростков, направленное на борьбу с их психологическими проблемами. Рассматриваются технические аспекты разработки такого функционала в приложение и приводится пример его реализации.

Ключевые слова: подростки, мобильное приложение, психологические проблемы, трекинг, рефлексия.

1. Введение

На сегодняшний день психологические проблемы у лиц подросткового возраста не являются чем-то экстраординарным. По данным Всемирной организации здравоохранения (ВОЗ), до двадцати процентов подростков в мире страдают от различных расстройств психики. В их число входят депрессия, тревожность и вызванные внешними факторами стрессы. Наиболее часто наблюдаются трудности в отношениях с родителями, переживания процесса учебной деятельности, зависимость от компьютерных игр и смартфонов, а также проблемы с самооценкой и принятием себя [1-2]. Эти проблемы оказывают негативное влияние на личность несовершеннолетних. Данная проблема требует своего решения самым комфортным для всех задействованных лиц способом. Около восьмидесяти процентов подростков имеют доступ к смартфону, что даёт понимание о потребности в мобильном приложении для получения квалифицированной помощи от специалистов.

2. Интеграция функционала “оценка дня” в мобильное приложение для подростков

В данной статье мы рассмотрим интеграцию функций трекинга изменения эмоционального состояния за день в мобильное приложение для подростков с психологическими проблемами, особо уделяя внимание функционалу "оценка дня". Данная функция позволяет пользователям оценивать свое эмоциональное состояние в течение дня, отслеживать его изменения и получать рекомендации от психологов в зависимости от полученных результатов. Для реализации данного функционала предлагается использовать алгоритмы машинного обучения, которые позволят анализировать данные о состоянии пользователя.

Рассмотрим реализацию описанной ранее функции трекинга на языке программирования Python. Во время создания кода использовалась библиотека машинного обучения Scikit-learn.

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Загрузка данных
data = pd.read_csv('emotional_state_data.csv')

# Обработка данных
X = data[['stress_level', 'activity_level',
          'mood']].values
y = data['emotional state'].values
```

```

# Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.2, random_state=42)

# Обучение алгоритма регрессии
model = LinearRegression()
model.fit(X_train, y_train)

# Тестирование алгоритма
y_pred = model.predict(X_test)

# Вывод результатов
print('Регрессионные коэффициенты:', model.coef_)
print('Среднеквадратичное отклонение:',
np.sqrt(np.mean((y_test - y_pred) ** 2)))

```

Рис. 1. Реализация функционала

Для контроля изменения эмоционального состояния за день необходимы данные об эмоциональном состоянии пользователя. Эти данные можно получить из различных источников, например: дневники эмоций, данные с мобильных устройств, таких как смарт-часы. Чтобы провести реализацию кода необходимо собирать данные о состоянии пользователя в течение дня, отправлять их на сервер для обработки и получать рекомендации в зависимости от полученных результатов.

Для начала, необходимо собрать данные о состоянии пользователя, которые могут включать в себя информацию о его настроении, уровне стресса, активности и других параметрах. Затем, эти данные необходимо обработать и преобразовать в формат, пригодный для анализа. Предлагается использовать методы нормализации и стандартизации данных, которые позволят сравнивать данные, полученные в разных единицах измерения. После необходимо разделить данные на обучающую и тестовую выборки. Обучающая выборка будет использоваться для обучения алгоритма машинного обучения, а тестовая выборка - для оценки качества работы алгоритма. Предлагается использовать алгоритм регрессии, который позволит нам предсказывать уровень эмоционального состояния пользователя на основе данных о его состоянии в течение дня.

В этом коде, мы загружаем данные о состоянии пользователя из файла csv, обрабатываем данные, разделяем их на обучающую и тестовую выборки, обучаем алгоритм регрессии и тестируем его на тестовой выборке. В результате, мы получаем регрессионные коэффициенты и среднеквадратичное отклонение, которое позволяет оценить качество работы алгоритма.

В целом, функционал "оценка дня" позволяет пользователям отслеживать изменения своего эмоционального состояния за день, получать рекомендации в зависимости от полученных результатов и улучшать свое общее психологическое здоровье. Реализация данного функционала на языке Python с использованием библиотеки Scikit-learn демонстрирует возможности использования алгоритмов машинного обучения для решения задач трекинга изменения эмоционального состояния за день в мобильном приложении для подростков с психологическими проблемами.

Литература

1. Жека Г.И., Кольчик Е.Ю. Психологические трудности в отношениях подростков с родителями // Теория и практика современной науки. 2018. №6 (36). С. 230-233.
2. Сидорова Т.А., Чернявская В.С. Групповое психологическое консультирование подростков с трудностями переживания процесса учебной деятельности // Педагогика и психология образования. 2022. №1. С. 254-265.

Численное моделирование модели Хиндмарша-Роуза

К.С. Рябов

Национальный исследовательский университет “Высшая Школа Экономики”

В данной научной работе исследуется феномен стохастической генерации квазипериодических осцилляций в трёхмерной модели нейрона Хиндмарш-Роуз. Автор изучает воздействие случайных возмущений на систему в зоне параметров, соответствующих периодическому осцилляционному режиму (спайкингу), и показывает, что при увеличении интенсивности шума этот режим трансформируется в квазипериодический (бёрстинг).

Ключевые слова: модель Хидмарша-Роуза, спайки, бёрсты.

1. Введение

Модель Хиндмарша-Роуза описывает генерацию бёрстов. Бёрсты представляют собой группы двух или более спайков, идущих подряд друг за другом и перемежаемых периодами отсутствия активности. Запишем систему уравнений данной модели и исследуем виды динамики, возникающие в ней при изменении значений параметров.

2. Описание модели

Математическая "нейронная" модель Хиндмарша-Роуза является одной из наиболее популярных моделей, поскольку качественно хорошо описывает динамику определённого класса нейронных моделей.

Предполагается, что нейронная модель, будучи нелинейной системой, может демонстрировать некоторые следующие типы активности нейронной клетки, включая состояние покоя, спайки и пачечную активность.

$$\begin{cases} \dot{x} = y - x^3 + bx^2 + I - z \\ \dot{y} = c - dx^2 - y \\ \dot{z} = r[s(x - x_1) - z] \end{cases}$$

Переменная x описывает мембранный потенциал нейрона, y и z соответствуют быстрому и медленному ионным токам, протекающим через мембрану, параметр $r \ll 1$ определяет отношение временных масштабов изменения этих токов, параметр I описывает внешний ток, подаваемый на нейрон, а остальные параметры характеризуют нелинейность проводимости мембраны.

3. Методы исследования

В данной работе реализовано численное моделирование системы Хиндмарша-Роуза при использовании параметра I в качестве управляющего параметра.

Параметры выберем следующие: $x_1 = -5/7, r = 0.001, s = 3, a = c = 1, b = 3, d = 5$.

Параметр I может варьироваться в диапазоне от -10 до 10 .

4. Результаты исследования

Приведем графики поведения траектории движение и зависимости мембранного потенциала при различных значениях параметра управления.

При $I = 2.2$:

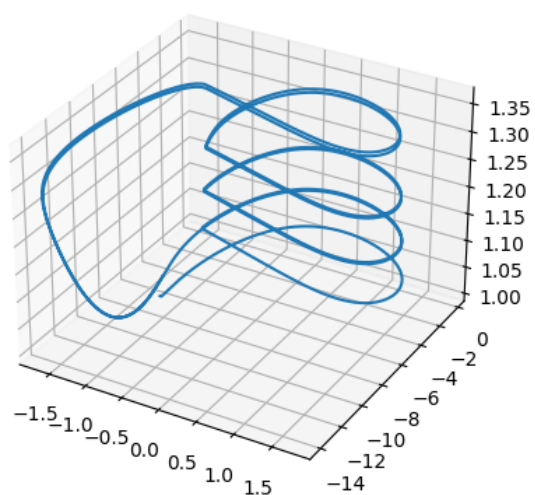


Рис. 1. Траектории движения при $I = 2.2$

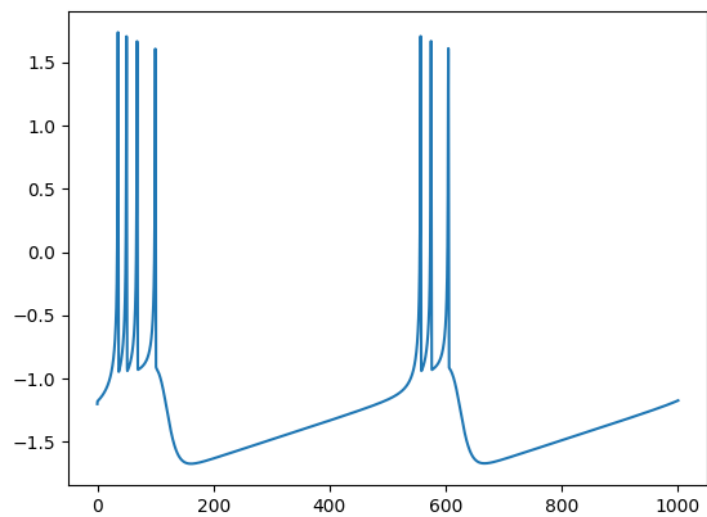


Рис. 2. Зависимость мембранного потенциала x от времени T при $I = 2.2$

При $I = 2.6$:

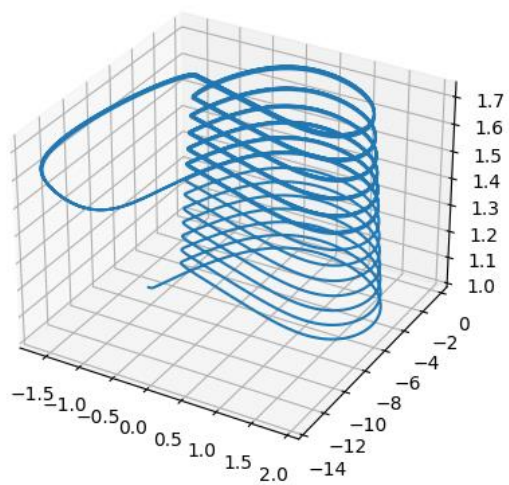


Рис. 3. Траектории движения при $I = 2.6$

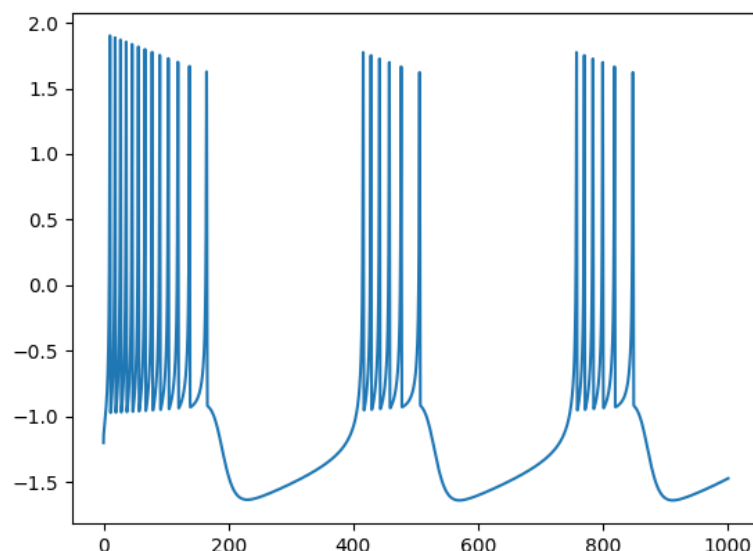


Рис. 4. Зависимость мембранного потенциала x от времени T при $I = 2.6$

На графиках траектории движения быстрая динамика характеризуется спайковыми частями, а медленная – покоящимися частями. На графиках зависимости мембранного потенциала быстрая динамика соответствует правой части аттрактора, то есть той части, где мы можем наблюдать спираль, а медленной динамике соответствует часть слева.

Спайки представлены генерацией потенциалов действия, а бёрсты – переходом мембранного потенциала из состояния покоя в состояние повторного возбуждения.

5. Выводы

Была рассмотрена модель Хиндмарша-Роуза с заданными значениями параметров. Благодаря малости параметра r в системе были выделены так называемые быстрые и медленные движения. В практической части исследования при помощи численного моделирования системы при изменении управляющего параметра I были построены траектории движения системы и графики зависимости мембранного потенциала x от времени. В результате были исследованы виды динамики, которые можно наблюдать в системе.

Система Хиндмарша-Роуза способна демонстрировать широкий спектр динамических режимов в зависимости от управляющих параметров. К числу этих режимов относятся регулярная и хаотическая генерации спайков или бёрстов, а также возбудимые режимы.

Литература

1. Некоркин В.И. Лекции по основам теории колебаний: Учебное пособие. – Нижний Новгород: Нижегородский госуниверситет, 2011. – 233с.
2. Ижикевич Е.М. Динамические системы в нейронауке. Геометрия возбудимости и пачечной активности. – М.–Ижевск: Ижевский институт компьютерных исследований, 2014. – 520 с.
3. Дмитричев А.С., Касаткин Д.В., Клиньшов В.В., Кириллов С.Ю., Масленников О. В., Щапин Д.С., Некоркин В.И. Нелинейные динамические модели нейронов: Обзор
4. Коломиец М., Шильников А. Методы качественной теории для модели Хиндмарш-Роуз // Нелинейная динамика, 2010. Т.6. № 1.
5. Chialvo D.R. Generic excitable dynamics on a two-dimensional map. Chaos, Solitons & Fractals, 1995, vol., No 3–4, p. 461.

Структура левого модуля планарной трехиндексной транспортной задачи

Е.Б. Титова

Нижегородский государственный университет им. Н.И. Лобачевского

Для матрицы ограничений планарной трехиндексной транспортной задачи рассмотрена структура левого нуль-пространства и построен базис модуля данного пространства.

Ключевые слова: планарная трехиндексная транспортная задача, левое нуль-пространство.

1. Введение

Пусть \mathbb{Q} — поле рациональных чисел, \mathbb{Q}^n — линейное пространство столбцов с компонентами из \mathbb{Q} , а \mathbb{Z}^n — столбцы из \mathbb{Q}^n с целочисленными компонентами.

Для $(m \times n)$ -матрицы A с элементами из \mathbb{Q} обозначим через $L_A = \{x \in \mathbb{Q}^m | xA = 0\}$ левое нуль-пространство матрицы A , а через L — какой-нибудь его базис.

Можно считать, что L — целочисленная $(m - r \times m)$ -матрица, где r — ранг матрицы A .

Множество $L_A^{\mathbb{Z}} = L_A \cap \mathbb{Z}^m$ назовем левым модулем матрицы A .

Множество целочисленных линейных комбинаций строк матрицы L содержится в $L_A^{\mathbb{Z}}$, но, вообще говоря, с ним не совпадает. Это выполняется тогда и только тогда, когда матрица L унимодулярна (т.е. НОД базисных миноров равен 1). В этом случае L будем называть базисом левого модуля матрицы A .

Обозначим через $M(A, b)$ множество рациональных решений системы $Ax = b$.

Базис L левого нуль-пространства L_A позволяет описать множество всех векторов $b \in \mathbb{Q}^m$, при которых $M(A, b) \neq \emptyset$.

По теореме Фредгольма [1] оно совпадает с множеством $\{b \in \mathbb{Q}^m | Lb = 0\}$.

Аналогичный вопрос над кольцом целых чисел \mathbb{Z} решается сложнее, однако имеет место следующий результат: если $b \in \mathbb{Z}^m$, A — унимодулярна и L — полная система векторов в L_A , то $M(A, b) \cap \mathbb{Z}^m \neq \emptyset$ тогда и только тогда, когда $Lb = 0$.

2. Постановка задачи

Рассмотрим ограничения-равенства планарной трехиндексной транспортной задачи (см., например, [2]) с неизвестными $x_{j_1 j_2 j_3} \geq 0$ ($j_i = 1, \dots, n_i$ при $i = 1, 2, 3$):

$$\begin{aligned} \sum_{j_1=0}^{n_1} x_{j_1 j_2 j_3} &= b_{0 j_2 j_3}, \\ \sum_{j_2=0}^{n_2} x_{j_1 j_2 j_3} &= b_{j_1 0 j_3}, \\ \sum_{j_3=0}^{n_3} x_{j_1 j_2 j_3} &= b_{j_1 j_2 0}, \end{aligned} \tag{1}$$

и обозначим через $T(n_1, n_2, n_3)$ её матрицу.

Из [3] следует, что матрицу $T(n_1, n_2, n_3)$ можно представить в виде строчечных блоков:

$$T(n_1, n_2, n_3) = \begin{pmatrix} I_{n_1} \times E_{n_2} \times E_{n_3} \\ E_{n_1} \times I_{n_2} \times E_{n_3} \\ E_{n_1} \times E_{n_2} \times I_{n_3} \end{pmatrix},$$

где $A \times B$ – кронекерово произведение [1] матриц A и B , E_n – единичная матрица n -го порядка, $I_n = (1, \dots, 1)$ – n -мерная строка, все компоненты которой равны 1.

Матрица $T(n_1, n_2, n_3)$ имеет число столбцов $N_T = n_1 n_2 n_3$, число строк $M_T = n_1 n_2 + n_1 n_3 + n_2 n_3$ и ранг $r_T = n_1 n_2 + n_1 n_3 + n_2 n_3 - n_1 - n_2 - n_3 + 1$. Левое нуль-пространства матрицы $T(n_1, n_2, n_3)$ имеет размерность $d_L(n_1, n_2, n_3) = M_T - r_T = n_1 + n_2 + n_3 - 1$.

Известно [1], что для совместности системы (1) необходимы так называемые условия баланса:

$$\begin{aligned} \sum_{j_1=0}^{n_1} b_{j_1 0 j_3} - \sum_{j_2=0}^{n_2} b_{0 j_2 j_3} &= 0, (j_3 = 1, \dots, n_3), \\ \sum_{j_1=0}^{n_1} b_{j_1 j_2 0} - \sum_{j_3=0}^{n_3} b_{0 j_2 j_3} &= 0, (j_2 = 1, \dots, n_2), \\ \sum_{j_2=0}^{n_2} b_{j_1 j_2 0} - \sum_{j_3=0}^{n_3} b_{j_1 0 j_3} &= 0, (j_1 = 1, \dots, n_1). \end{aligned} \quad (2)$$

3. Основные результаты

В [4] дано описание построения левого нуль-пространства. Следуя ему, условия баланса (2) можно представить в блочном виде:

$$L(n_1, n_2, n_3) = \begin{pmatrix} I_1 \times I_{n_2} \times E_{n_3} & -I_{n_1} \times I_1 \times E_{n_3} & 0_{n_3 \times n_1 n_2} \\ I_1 \times E_{n_2} \times I_{n_3} & 0_{n_2 \times n_1 n_3} & -I_{n_1} \times E_{n_2} \times I_1 \\ 0_{n_1 \times n_2 n_3} & E_{n_1} \times I_1 \times I_{n_3} & -E_{n_1} \times I_{n_2} \times I_1 \end{pmatrix},$$

где $0_{m \times n}$ – нулевая $(m \times n)$ -матрица.

Непосредственно перемножая матрицы, получаем, что $L(n_1, n_2, n_3) \cdot T(n_1, n_2, n_3) = 0$.

Матрица $L(n_1, n_2, n_3)$ имеет число строк $M_L = n_1 + n_2 + n_3$, число столбцов $N_L = n_1 n_2 + n_1 n_3 + n_2 n_3$.

Лемма 1. Матрица $L(n_1, n_2, n_3)$ вполне унимодулярна и имеет ранг, равный $d_L(n_1, n_2, n_3)$.

Доказательство. Матрица $L(n_1, n_2, n_3)$ представляет собой матрицу инцидентности полного ориентированного графа, следовательно, вполне унимодулярна и имеет ранг $r_L = M_L - 1$, что совпадает с размерностью $d_L(n_1, n_2, n_3)$ левого нуль-пространства матрицы $T(n_1, n_2, n_3)$. ■

Следствие 1. Для построения базиса пространства L_T^Z левого модуля матрицы $T(n_1, n_2, n_3)$ достаточно удалить любую из строк матрицы $L(n_1, n_2, n_3)$.

4. Заключение

В данной работе левое нуль-пространства трехиндексной планарной транспортной задачи задано с помощью кронекерова произведения матриц и получен базис модуля данного пространства, что дает ключ к дальнейшему исследованию.

Литература

1. Воеводин В.В., Кузнецов Ю.А. СМБ. Матрицы и вычисления. М.: Наука, 1984.
2. Емеличев В.А., Ковалёв М.М., Кравцов М.К. Многогранники, графы, оптимизация. М.: Наука, 1981.
3. Шевченко В.Н. Характеристические многочлены многоиндексных транспортных задач. //Дискретная математика Т.15. Вып. 2. 2003. С. 83–88.
4. Шевченко В.Н. Многогранники многоиндексных транспортных задач: алгебраический подход. //Материалы конференции «Дискретный анализ и исследование операций» (Новосибирск, 28 июня – 2 июля 2004). – Новосибирск: Изд-во ин-та математики, 2004. С. 64–70.

Критерий подобия над кольцом целых чисел нильпотентных верхних треугольных матриц 4-го порядка матрицам с одной супердиагональю

Г.В. Уткин

Нижегородский государственный университет им. Н.И. Лобачевского

Работа посвящена проблеме распознавания подобия над кольцом целых чисел нильпотентных верхних треугольных матриц 4-го порядка максимального ранга и матриц с одной супердиагональю. Получены необходимые и достаточные условия подобия таких матриц в простых терминах делимости и наибольших общих делителей элементов матриц.

Ключевые слова: подобие матриц, жорданова клетка, обобщение жордановой клетки, нильпотентная матрица, кольцо целых чисел.

1. Введение

Из линейной алгебры известно (см., например, [1]), что любая нильпотентная матрица порядка n ранга $n - 1$ подобна над полем \mathbb{Q} жордановой клетке J_n

$$J_n = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Над кольцом \mathbb{Z} этот факт неверен (см [2–8]). К примеру, любая матрица $A = \begin{pmatrix} 0 & a \\ 0 & 0 \end{pmatrix}$, где $a \neq 0$, подобна над \mathbb{Q} матрице J_2 . Если $a \notin \{0, 1, -1\}$, то над кольцом \mathbb{Z} такая матрица A не подобна жордановой клетке J_2 .

Обобщением жордановой клетки назовём матрицу вида

$$\text{superdiag}(a_1, a_2, \dots, a_{n-1}) = \begin{pmatrix} 0 & a_1 & 0 & \cdots & 0 \\ 0 & 0 & a_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-1} \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Интересным является вопрос, при каких условиях нильпотентная матрица порядка n ранга $n - 1$ подобна над \mathbb{Z} обобщению жордановой клетки. Случай $n = 2$ тривиален. Для $n = 3$ критерий получен в [3, 4, 6]. В данной заметке мы ограничимся случаем $n = 4$ и установим необходимые и достаточные условия, при которых верхняя треугольная нильпотентная матрица 4-го порядка и ранга 3 подобна обобщению жордановой клетки. Напомним необходимые определения.

Определение 1. Будем говорить, что матрица $A \in \mathbb{Z}^{n \times n}$ подобна матрице $B \in \mathbb{Z}^{n \times n}$ над кольцом целых чисел \mathbb{Z} , если существует такая унимодулярная матрица $X \in GL(n, \mathbb{Z})$, что $AX = XB$. При этом матрицу X называют трансформирующей.

Факт подобия матриц A и B над кольцом целых чисел \mathbb{Z} будем обозначать $A \sim B$.

Известно (см. [3, 5]), что если A и B – две нильпотентные верхние треугольные матрицы порядка n , причём все элементы их первых супердиагоналей отличны от 0 (т.е. $\text{rank } A = \text{rank } B = n - 1$), и $A \sim B$, то существует унитарная трансформирующая матрица X (т.е. $AX = XB$ и X – верхняя треугольная матрица с единичной диагональю).

2. Критерий подобия матриц

В этом разделе мы получим критерий подобия над кольцом целых чисел нильпотентных матриц следующего вида:

$$A = \begin{pmatrix} 0 & a_1 & b_1 & c_1 \\ 0 & 0 & a_2 & b_2 \\ 0 & 0 & 0 & a_3 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & a_1 & 0 & 0 \\ 0 & 0 & a_2 & 0 \\ 0 & 0 & 0 & a_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \text{superdiag}(a_1, a_2, a_3), \quad (1)$$

где $a_i \neq 0$ ($i = 1, 2, 3$).

Как уже было отмечено выше, если матрицы A и B вида (1) подобны над \mathbb{Z} , то существует унитарная трансформирующая матрица

$$X = \begin{pmatrix} 1 & x_1 & y_1 & z_1 \\ 0 & 1 & x_2 & y_2 \\ 0 & 0 & 1 & x_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2)$$

Поскольку для такой матрицы X условие $X \in GL(n, \mathbb{Z})$ автоматически выполняется, то равенство $AX = XB$ превращается в систему линейных уравнений над \mathbb{Z} . Таким образом, $A \sim B$ тогда и только тогда, когда эта система имеет решение в целых числах. Известен следующий критерий совместности в целых числах системы линейных уравнений.

Теорема 1. (см. [9, с. 51]). Пусть $P \in \mathbb{Z}^{m \times n}$ – матрица ранга m и $b \in \mathbb{Z}^m$. Тогда система $Px = b$ имеет целочисленные решения тогда и только тогда, когда все миноры порядка m расширенной матрицы (P, b) делятся на наибольший общий делитель $\Delta_m(P)$ всех миноров порядка m матрицы P .

В следующей теореме устанавливается критерий подобия над \mathbb{Z} матриц вида (1).

Теорема 2. Рассмотрим матрицы вида (1). Пусть $a_1, a_2, a_3 \neq 0$ и $\text{НОД}(a_1, a_2, a_3) = 1$. Тогда $A \sim B$ тогда и только тогда, когда выполняются следующие условия:

- 1) $(a_3 b_1 + a_1 b_2) : a_2$,
- 2) $(b_1 b_2 - a_2 c_1) : \text{НОД}(a_1, a_3)$.

Доказательство. Поскольку трансформирующая матрица X имеет вид (2), то условие $AX = XB$ равносильно системе линейных уравнений:

$$\begin{cases} b_1 + a_1 x_2 = a_2 x_1, \\ b_2 + a_2 x_3 = a_3 x_2, \\ c_1 + b_1 x_3 + a_1 y_2 = a_3 y_1. \end{cases}$$

Выпишем матрицу этой системы:

$$\left(\begin{array}{cccccc|c} x_1 & x_2 & x_3 & y_1 & y_2 & z_1 & \\ a_2 & -a_1 & 0 & 0 & 0 & 0 & b_1 \\ 0 & a_3 & -a_2 & 0 & 0 & 0 & b_2 \\ 0 & 0 & -b_1 & a_3 & -a_1 & 0 & c_1 \end{array} \right)$$

Выясним, при каких условиях эта система имеет целочисленное решение. Согласно Теореме 1 данная система совместна над \mathbb{Z} тогда и только тогда, когда все миноры 3-го порядка её расширенной матрицы делятся на Δ_3 . Имеем

$$\begin{aligned}\Delta_3 &= \text{НОД}(-a_2 a_3 b_1, a_2 a_3 a_3, -a_1 a_2 a_3, -a_2 a_2 a_3, a_1 a_2 a_2, a_1 a_2 a_3, -a_1 a_1 a_2) = \\ &= a_2 \cdot \text{НОД}(a_3 b_1, a_3 a_3, a_1 a_3, a_2 a_3, a_1 a_2, a_1 a_1) = \\ &= a_2 \cdot \text{НОД}(a_3 \cdot \text{НОД}(b_1, a_1, a_2, a_3), a_1 a_2, a_1 a_1) = \\ &= a_2 \cdot \text{НОД}(a_3, a_1 \cdot \text{НОД}(a_1, a_2)) = \\ &= a_2 \cdot \text{НОД}(a_1, a_3).\end{aligned}$$

Выпишем миноры 3-го порядка расширенной матрицы, отличные от миноров основной матрицы:

$$\begin{aligned}M_1 &= a_2 a_3 c_1, M_2 = a_2(b_1 b_2 - a_2 c_1), M_3 = -a_2 a_3 b_2, M_4 = a_1 a_2 b_2, \\ M_5 &= -b_1(a_3 b_1 + a_1 b_2) + a_1 a_2 c_1, M_6 = a_3(a_3 b_1 + a_1 b_2), M_7 = -a_1(a_3 b_1 + a_1 b_2), \\ M_8 &= -a_2 a_3 b_1, M_9 = a_1 a_2 b_1.\end{aligned}$$

Заметим, что миноры M_1, M_3, M_4, M_8 и M_9 заведомо делятся на Δ_3 . Минор M_2 делится на Δ_3 тогда и только тогда, когда $(b_1 b_2 - a_2 c_1) : \text{НОД}(a_1, a_3)$. Обозначив $s = a_3 b_1 + a_1 b_2$, имеем $M_5 = -b_1 s + a_1 a_2 c_1, M_6 = a_3 s, M_7 = -a_1 s$. Поскольку $a_1 a_2 : \Delta_3$, то M_5, M_6, M_7 кратны Δ_3 тогда и только тогда, когда $b_1 s, a_3 s, a_1 s : \Delta_3$, что в свою очередь равносильно условию

$$\text{НОД}(a_1 s, a_3 s, b_1 s) = s \cdot \text{НОД}(a_1, a_3, b_1) \text{ делится на } (a_2 \cdot \text{НОД}(a_1, a_3)).$$

Это означает, что

$$s \text{ делится на } a_2 \cdot \frac{\text{НОД}(a_1, a_3)}{\text{НОД}(a_1, a_3, b_1)}. \quad (3)$$

Поскольку $\text{НОД}(a_1, a_2, a_3) = 1$, то сомножители a_2 и $\frac{\text{НОД}(a_1, a_3)}{\text{НОД}(a_1, a_3, b_1)}$ взаимно простые. Следовательно, свойство (3) равносильно тому, что $s = (a_3 b_1 + a_1 b_2) : a_2$. ■

Следствие 1. Рассмотрим матрицы вида (1). Пусть $a_1, a_2, a_3 \neq 0$ и $d = \text{НОД}(a_1, a_2, a_3)$. Тогда $A \sim B$ тогда и только тогда, когда

- 1) $b_1 : d, b_2 : d, c_1 : d$,
- 2) $(b_1 b_2 - a_2 c_1) : (d \cdot \text{НОД}(a_1, a_3))$,
- 3) $(a_3 b_1 + a_1 b_2) : d a_2$.

Доказательство. Пусть $d = \text{НОД}(a_1, a_2, a_3)$. Условие $A \sim B$ равносильно тому, что $c_i, b_i : d, i = 1, 2$ и $\frac{1}{d} A \sim \frac{1}{d} B$. Поскольку матрицы $\frac{1}{d} A$ и $\frac{1}{d} B$ удовлетворяют условиям Теоремы 2, то $\frac{1}{d} A \sim \frac{1}{d} B$ тогда и только тогда, когда

$$\left(\frac{a_3 b_1}{d} + \frac{a_1 b_2}{d}\right) : \frac{a_2}{d} \text{ и } \left(\frac{b_1 b_2}{d} - \frac{a_2 c_1}{d}\right) : \text{НОД}\left(\frac{a_1}{d}, \frac{a_3}{d}\right).$$

Это равносильно тому, что $(a_3 b_1 + a_1 b_2) : d a_2$ и $(b_1 b_2 - a_2 c_1) : (d \cdot \text{НОД}(a_1, a_3))$. С учётом условия $c_i, b_i : d, i = 1, 2$ получаем требуемое. ■

Следствие 2. Рассмотрим матрицы вида (1). Пусть $a_1, a_2, a_3 \neq 0$ и $\text{НОД}(a_1, a_3) = 1$. Тогда

$$A \sim B \Leftrightarrow (a_3 b_1 + a_1 b_2) : a_2.$$

Отметим, что доказанная Теорема 2 является обобщением Теоремы 4.2 из [4].

Благодарности

Автор выражает особую благодарность Сидорову Сергею Владимировичу за полезные советы, важные замечания, постоянное внимание к работе и рекомендации по оформлению.

Литература

1. Гантмахер Ф.Р. Теория матриц. – М. Наука, 1988. – С. 552.
2. Huser D. Similarity of integer matrices, PhD Thesis, University of Paderborn, 2017.
3. Appelgate H., Onishi H. The similarity problem 3×3 integer matrices // Linear Algebra Appl. 1982. Vol 42. P. 159–174.
4. Сидоров С.В., Уткин Г.В. О подобии над кольцом целых чисел некоторых нильпотентных матриц максимального ранга // Журнал СВМО. 2023. Т. 25, № 4. С. 284–298.
5. Сидоров С.В. О подобии некоторых целочисленных матриц с единственным собственным значением над кольцом целых чисел // Матем. заметки. 2019. Т. 105, № 5. С. 763–770.
6. Сидоров С.В. О подобии матриц третьего порядка над кольцом целых чисел, имеющих приводимый характеристический многочлен // Вестник Нижегородского университета им. Н.И. Лобачевского. 2009. № 1. С. 119–127
7. Сидоров С.В. Выделение эффективно разрешимых классов в задаче подобия матриц над кольцом целых чисел, дис. канд. физ.-мат. наук: 01.01.09: защищена 15.10.15: утв. 21.03.16, Нижний Новгород, 2015.
8. Шевченко В.Н., Сидоров С.В. О подобии матриц второго порядка над кольцом целых чисел // Известия высших учебных заведений. Математика. 2006. № 4. С. 57– 64.
9. Schrijver A. Theory of Linear and Integer Programming. Wiley, 1998.

Алгоритм размытия границ изображения на основе нечетких множеств

Д.Е. Фомичев, А.Ф. Ляхов

Нижегородский государственный университет им. Н.И. Лобачевского

В статье описан алгоритм размытия границ изображения на основе нечетких множеств. Применение данного алгоритма к простым контурам, таким как отрезок, квадрат и круг. Исследование способов суммирования функций принадлежности, проверка совпадения этих методов и их применение к размытию границ.

Ключевые слова: нечеткие множества, размытие границ объектов, распознавание образов, обработка изображений.

1. Введение

Распознавание образов изображения является основной проблемой классификации объектов, и она лежит в основе программ управления, относящихся к сфере искусственного интеллекта.

Большинство методов распознавания объектов связаны с созданием вектора признаков объекта. Как правило, некоторые элементы этого вектора определяются размытой границей объекта. Эффект размытости возникает при движении объекта или при его маскировке. В настоящее время, разработано много методов, которые нечёткую границу объекта переводят в чёткую, резкую границу [1,3]. Однако существует ряд практических задач, к которым не применим данный подход. Например, требуется обнаружить след, оставляемый катером на поверхности воды, который с течением времени размывается. Эта задача ещё больше усложняется при учёте случайного поверхностного волнения.

Можно предложить новый подход к созданию вектора признаков включающие размытую границу и её параметры размытости. Изменяя параметры размытия можно получить визуально различные картины объекта. Этот параметр может стать одной из компонент вектора признаков.

В работе предлагается метод размытия чёткой границы изображения и определяется метод введения функций количественно описывающих размытость.

2. Размытие чётких границ изображений

Рассмотрим растровое изображение – двумерный массив пикселей. Каждый пиксель представляется кортежем чисел, определяемый цветовой моделью. В работе рассматривается цветовая модель RGB (red, green, blue), где каждая компонента принимает значения от 0 до 255.

2.1 Алгоритм размытия границ

Пусть изображение — тонкая чёрно-белая полоска высотой в 1 пиксель и шириной в a пикселей (Рис. 1).

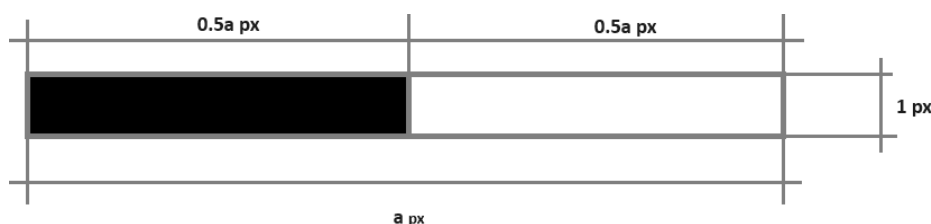


Рис. 1. Тонкая чёрно-белая полоска

Координаты всех пикселей данного изображения образуют чёткое множество:

$$L = \{x \in N \cup \{0\}: 0 \leq x \leq a\}.$$

Чёрному цвету будут соответствовать значения пикселей от 0 до $\frac{a}{2}$, а белому от $\frac{a}{2} + 1$ до a . Таким образом, определяются два четких подмножества. Координаты пикселей, соответствующие черному цвету, образуют подмножество:

$$B = \{x \in N \cup \{0\}: 0 \leq x \leq \frac{a}{2}\},$$

координаты белого цвета:

$$W = \{x \in N \cup \{0\}: \frac{a}{2} + 1 \leq x \leq a\}.$$

Для фаззификации множеств необходимо определить коэффициенты принадлежности к нечёткому множеству [2]. Для расчёта коэффициентов задаётся функция принадлежности: $\mu_A(u, t, n, k)$, где A — нечеткое множество, u — элемент универсального множества, t — левая граница, n — середина, а k — правая граница множества A .

Для подмножества B функция принадлежности задаётся в виде:

$$\mu_B = \mu_B\left(x, 0, \frac{a}{4}, \frac{a}{2}\right),$$

где $x \in L$.

Для подмножества W :

$$\mu_W = \mu_W\left(x, \frac{a}{2} + 1, \frac{3a}{4}, a\right).$$

В результате, получаются два нечётких множества: $\{(\mu_B, x): x \in L\}$ и $\{(\mu_W, x): x \in L\}$. После объединения этих множеств получится новое:

$$\{(max(\mu_B, \mu_W), x): x \in L\}.$$

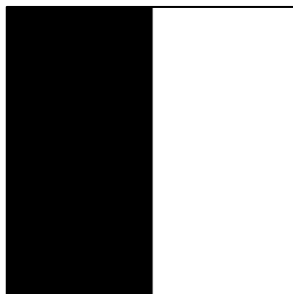
Теперь каждому пикселю исходного изображения соответствует коэффициент принадлежности. Для фаззификации необходимо умножить коэффициент на значение цвета каждого пикселя. Множество L назовем отрезком размытия.

Для исследования всех задач в статье использовалась программа Fuzzy Logic Graphic Instruments, разработанная для обработки изображений с помощью аппарата нечеткой логики.

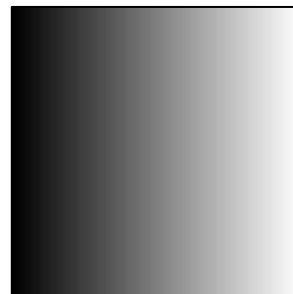
2.2 Размытие по отрезку

2.2.1 Размытие по горизонтали

Даны две вертикальные полосы чёрного и белого цветов (Рис. 2(а)). Множество координат пикселей изображения обозначим как L .



а) Вертикальные полосы до размытия



б) Вертикальные полосы после размытия

Рис. 2. Применение алгоритма размытия с треугольной функцией принадлежности для вертикальных полос

Отрезок размытия образует множество:

$$L = \left\{ (x, y) \in I : \begin{matrix} 0 \leq x \leq a \\ \forall y \end{matrix} \right\},$$

где a – ширина изображения.

Для демонстрации работы алгоритма используется треугольная функция принадлежности. В общем виде она выглядит так:

$$\mu_G(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b < x \leq c \\ 0, & x > c \end{cases},$$

где a – левая граница размытия, b – середина, а c – правая граница. G – нечеткое множество.

После применения алгоритма размытия границ в каждой точке отрезка будет вычислен коэффициент размытия.

Таким образом, каждая точка отрезка L – абсцисса координат пикселя исходного изображения. Необходимо умножить коэффициент размытия, соответствующей абсциссы, на каждый пиксель по вертикали. В результате получится размытая граница (Рис. 2(b)).

2.2.2 Размытие по вертикали

Даны две вертикальные полосы черного и белого цветов (Рис. 3(a)). Множество координат пикселей изображения обозначим как I .

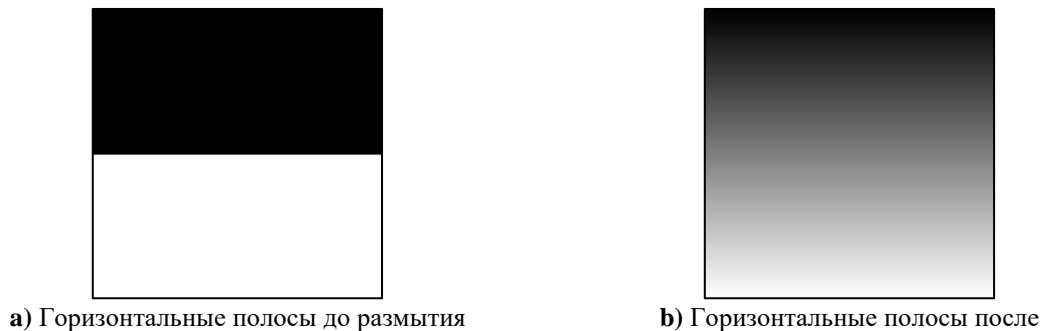


Рис. 3. Применение алгоритма размытия с треугольной функцией принадлежности для горизонтальных полос

Отрезок размытия образует множество:

$$L = \left\{ (x, y) \in I : \begin{matrix} \forall x \\ 0 \leq y \leq a \end{matrix} \right\},$$

где a – высота изображения.

Для фаззификации будем использовать треугольную функцию принадлежности. После применения алгоритма размытия границ в каждой точке отрезка будет вычислен коэффициент размытия. Таким образом, каждая точка отрезка L – ордината координат пикселя исходного изображения. Необходимо умножить коэффициент размытия, соответствующей ординаты, на каждый пиксель по горизонтали. В результате получится размытая граница (Рис. 3(b)).

2.3 Размытие простых фигур

2.3.1 Размытие границ прямоугольника

Дано изображение прямоугольника (Рис. 5(а)). I – множество всех координат пикселей изображения. Координаты пикселей, образующих прямоугольник, образуют множество:

$$P = \left\{ (x, y) \in I : \begin{matrix} a \leq x \leq b \\ c \leq y \leq d \end{matrix} \right\}.$$

Точка пересечения диагоналей – $O\left(\frac{a+b}{2}, \frac{c+d}{2}\right)$. Длина диагонали по x : $d_x = b - \frac{a+b}{2}$. Длина диагонали по y : $d_y = d - \frac{c+d}{2}$.

Коэффициентом степени размытия определим величину $k = [0,1]$, отвечающую за интенсивность размытия границ.

Длина размытия по x : $\delta_x = kd_x$. Длина размытия по y : $\delta_y = kd_y$.

Отрезком размытия определим множество точек:

$$L = \left\{ (x, y) \in I : \begin{matrix} \frac{a+b}{2} + \delta_x \leq x \leq b + \delta_x \\ \frac{c+d}{2} + \delta_y \leq y \leq d + \delta_y \end{matrix} \right\}.$$

После задания отрезка размытия применим алгоритм размытия границ. Выберем треугольную функцию принадлежности. Для обхода точек разобьём исходный прямоугольник на периметры меньших

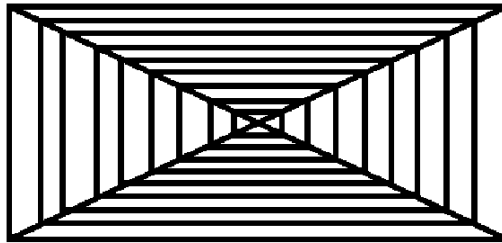
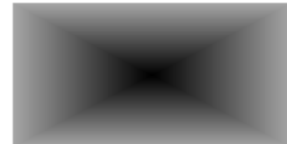
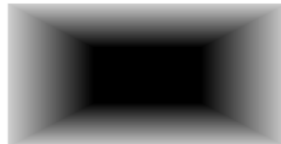


Рис. 4. Разбиение прямоугольника на периметры меньших

Каждому периметру будет принадлежать точка из отрезка размытия и соответствующий цвет. Необходимо пройти по всем периметрам и заменить на новый цвет.

Результат работы алгоритма с треугольной функцией принадлежности с коэффициентом степени размытия $k = 0.5$ изображён на рис. 5(б), с $k = 1$ – на рис. 5(с).



а) Прямоугольник до размытия б) Размытие границ с коэффициентом степени размытия $k=0.5$ в) Размытие границ с коэффициентом степени размытия $k=1$

Рис. 5. Применение алгоритма размытия с треугольной функцией принадлежности для прямоугольника

2.3.2 Размытие границ круга

Дано изображение круга (Рис. 7(a)). Множество всех координат пикселей изображения обозначим как I . Координаты пикселей, образующих круг, образуют множество:

$$K = \{(x, y) \in I: (x - x_0)^2 + (y - y_0)^2 \leq R^2\},$$

где R – радиус граничной окружности, $O(x_0, y_0)$ – центр окружности.

Коэффициентом степени размытия определим величину $k = [0,1]$, отвечающую за интенсивность размытия границ. Тогда длина размытия: $r = kR$.

Отрезком размытия определим множество:

$$L = \left\{ (x, y) \in I: \begin{matrix} R - r \leq x \leq R + r \\ \forall y \end{matrix} \right\}.$$

После задания отрезка размытия применим алгоритм размытия границ. Функция принадлежности будет треугольной. Для обхода точек разобьём круг на окружности меньших радиусов (Рис. 6).

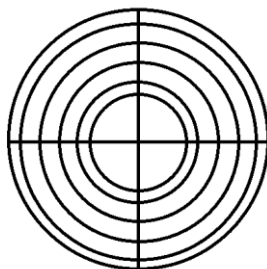


Рис. 6. Разбиение круга на окружности меньших радиусов

Обход внутренних окружностей необходимо выполнить с учётом параметризации:

$$\begin{cases} x = a \cos(\varphi) + x_0 \\ y = a \sin(\varphi) + y_0 \end{cases},$$

где a — радиус текущей окружности, $\varphi \in [0, 2\pi]$.

Каждой окружности будет соответствовать новый цвет. Необходимо пройти по всем окружностям и заменить на новый цвет.

Результат работы алгоритма с треугольной функцией принадлежности с коэффициентом степени размытия $k = 0.5$ изображен на рис. 7(b), с $k = 1$ – на рис. 7(c).

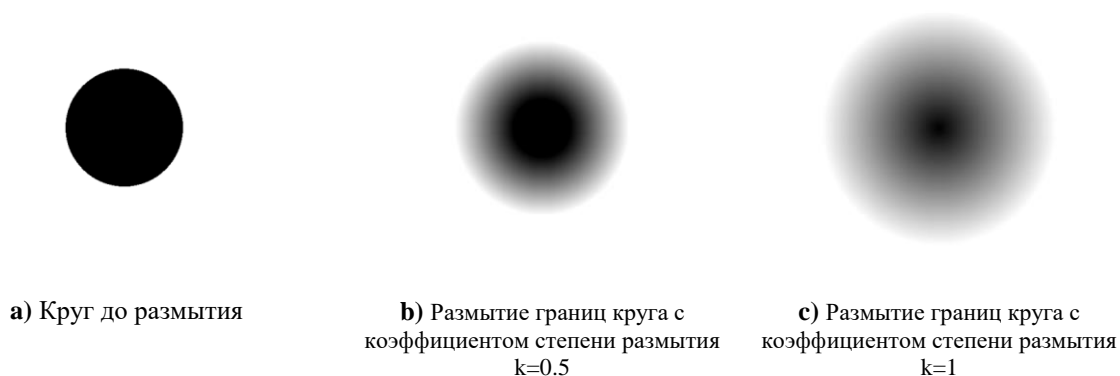


Рис. 7. Применение алгоритма размытия для круга

3. Размытие с несколькими параметрами

На практике размытость границы редко определяется одной функцией принадлежности. Для моделирования размытия может понадобиться комбинация функций принадлежности. Одним из способов суммирования является получение функции принадлежности как объединение начальных функций принадлежности. Другой подход основан на нахождение среднего арифметического между коэффициентами принадлежности.

Покажем, что, эти способы приводят к различным видам размытия. Размытие, получаемое при объединении функций принадлежности, не равно сумме размытий, полученных отдельными функциями. В качестве суммируемых функций принадлежности рассмотрим треугольную и экспоненциальную с показателем в квадрате.

3.1 Треугольная функция принадлежности

Треугольная функция принадлежности определяется как:

$$\mu_G(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b < x \leq c \\ 0, & x > c \end{cases},$$

где a – левая граница размытия, b – середина, c – правая граница. G – нечеткое множество.

3.2 Экспоненциальная функция принадлежности с показателем в квадрате

Экспоненциальная функция принадлежности с показателем в квадрате определяется как:

$$\mu_G(x) = \begin{cases} 0, & x < a \\ \exp(-(x-b)^2), & a \leq x \leq b \\ \exp(-(x-b)^2), & b < x \leq c \\ 0, & x > c \end{cases},$$

где a – левая граница размытия, b – середина, c – правая граница, G – нечеткое множество.

3.3 Комбинация функций принадлежности

Обозначим треугольную функцию принадлежности $\mu_{1G}(x)$, экспоненциальную с показателем в квадрате соответственно $\mu_{2G}(x)$.

Нахождение среднего арифметическое из двух функций принадлежности:

$$\mu_{3G}(x) = \frac{\mu_{1G}(x) + \mu_{2G}(x)}{2}.$$

Объединение двух функций принадлежности определяется как:

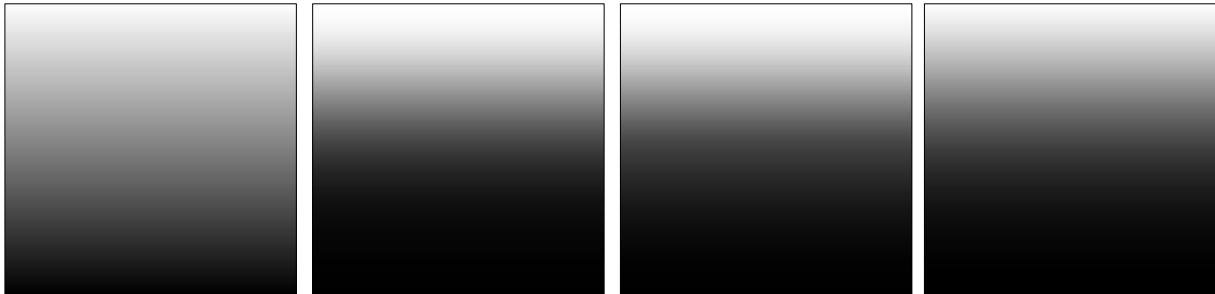
$$\mu_{4G}(x) = \max\{\mu_{1G}(x), \mu_{2G}(x)\}.$$

Результат применения алгоритма размытия по вертикали к белой и черной горизонтальным полосам показан на рис. 8(a), с функциями принадлежности: $\mu_{1G}(x)$ – на рис.8(b), $\mu_{2G}(x)$ – на рис.8(c), $\mu_{3G}(x)$ – на рис. 8(d), $\mu_{4G}(x)$ – на рис. 8(e).

Непосредственное визуальное сравнение полученных размытых границ (см. рис.8(d) и рис.8(e)) не позволяет количественно оценить близость или совпадение $\mu_{3G}(x)$ и $\mu_{4G}(x)$.



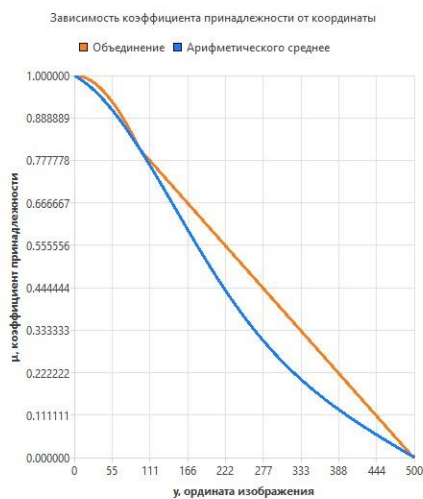
а) Исходное изображение: белая и черная горизонтальные полосы



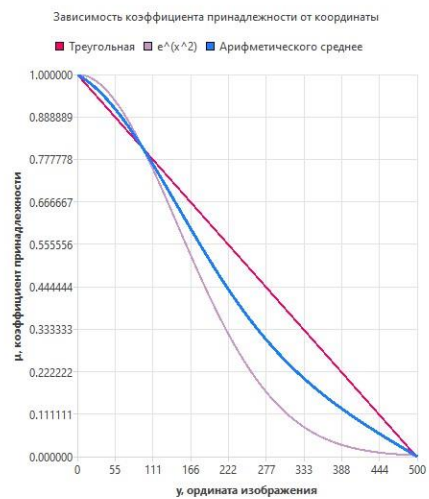
- б) Размытие по вертикали с треугольной функцией принадлежности** **с) Размытие по вертикали с экспоненциальной функцией принадлежности с показателем в квадрате** **д) Объединение треугольной и экспоненциальной с показателем в квадрате функций принадлежности** **е) Среднее арифметическое из треугольной и экспоненциальной с показателем в квадрате функций принадлежности**

Рис. 8. Применение различных функций принадлежности к изображению

Для получения количественной оценки введём функцию зависимости коэффициента принадлежности от ординаты изображения. Графики этих функций показаны на рис. 9.

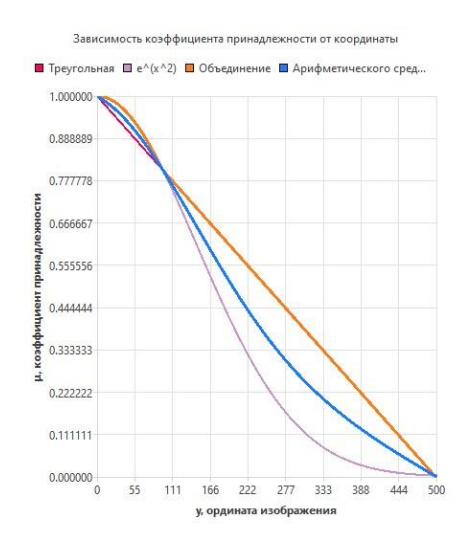


а) Сравнение суммирования с помощью объединения и арифметического среднего

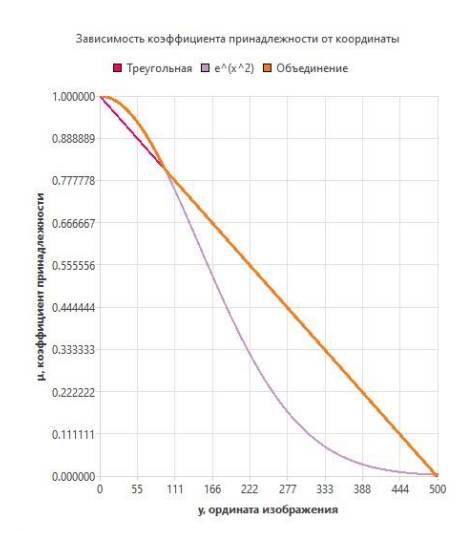


б) Сравнение треугольной, экспоненциальной с показателем в квадрате и их арифметического среднего

Рис. 9. Графики зависимости коэффициента принадлежности от ординаты изображения



с) Сравнение всех функций принадлежности: треугольная, экспоненциальная в квадрате, объединения, арифметического среднего



д) Сравнение треугольной, экспоненциальной с показателем в квадрате и объединения функций принадлежности

Рис. 9. Графики зависимости коэффициента принадлежности от ординаты изображения

Сравнивая поведение кривой арифметического среднего от треугольной и экспоненциальной с показателем в квадрате функций принадлежности и кривой их объединения (Рис. 9(а)), можно сделать вывод, что размытые границы, полученные этими способами, различны.

4. Заключение

В работе предложен и реализован в программе Fuzzy Logic Graphic Instruments алгоритм размытия границ изображения. С его помощью реализовано размытие по вертикали и горизонтали, а также простых фигур – прямоугольника и круга. Сделан вывод об отсутствии полного совпадения размытия границ при разных способах суммирования функций принадлежности.

Различные способы суммирования функций принадлежности позволяют использовать их для моделирования размытия границ реальных изображений и формирования элементов вектора признаков объекта.

Литература

1. Ляхов А.Ф. Аппроксимация функции с помощью нечётких множеств в пакете FUZZY LOGIC TOOLBOX. – Нижегородский государственный университет им. Н.И. Лобачевского. – 4 с.
2. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику. -М.: Горячая линия – Телеком, 2006. – 212 с.
3. Потапов А.А., Гуляев Ю.В., Никитов С.А., Пахомов А.А., Герман В.А. Новейшие методы обработки изображений. / Под ред. А.А. Потапова. — М.: ФИЗМАТЛИТ, 2008. — 496 с.

Содержание

Предисловие	3
В.А. Бородина, О.А. Монахова	
Подвижная конструкция, составленная из замечательных математических кривых	5
Г.И. Захарычев	
Реализация полиномиальной и сплайн интерполяции на Python	9
О.А. Зорин, Е.А. Козин	
Распределённая версия алгоритма глобального поиска в фреймворке интеллектуальной оптимизации «iOpt»	14
П.А. Кочетков	
Разработка обучающего интерактивного онлайн портала по дисциплине «Методы оптимизации»	20
В.В. Кузнецов, Н.Ю. Золотых	
Диагностика сердечно-сосудистых заболеваний с использованием сверточной нейронной сети	24
Е.С. Пинежанин, К.А. Баркалов	
Об исследовании нового класса модельных задач для проведения экспериментов с методами условной глобальной оптимизации	33
V.V. Razin, A.A. Krasnov	
Determining the presence of heart disease by ECG in the PTB-XL dataset using machine learning ...	38
V.V. Razin, A.A. Krasnov	
Solving the problem of diagnosing a disease by ECG on the PTB-XL dataset using machine learning.....	43
А.Р. Резникова	
Разработка функции для трекинга эмоций в мобильном приложении для подростков с психологическими проблемами	49
К.С. Рябов	
Численное моделирование модели Хиндмарша-Роуза	51
Е.Б. Титова	
Структура левого модуля планарной трехиндексной транспортной задачи	54
Г.В. Уткин	
Критерий подобия над кольцом целых чисел нильпотентных верхних треугольных матриц 4-го порядка матрицам с одной супердиагональю.....	57

Д.Е. Фомичев, А.Ф. Ляхов

Алгоритм размытия границ изображения на основе нечетких множеств 61

Математика и ИТ – вместе в цифровое будущее

Сборник трудов III Молодежной школы

Нижний Новгород, 22–26 апреля 2024 г.

Под ред. К.А. Баркалова, М.А. Богомоловой

Отв. за выпуск Т.Г. Смирнова

Публикуется в авторской редакции

ННГУ им. Н.И. Лобачевского.
603022, Н. Новгород, пр. Гагарина, 23.