Лекция 1. Основные термины в машинном обучении. Базовая математика в машинном обучении.

Глинский А.В.

Цель занятия: знакомство с основными терминами, а также с базовым математическим аппаратом, используемым в машинном обучении

Содержание

0.1 Вводная информация о курсе

Здравствуйте, сегодня мы начинаем изучение курса «Математика и алгоритмы машинного обучения». Замечу, что необходимым условием для успешного прохождения курса является базовое знание языка python и умение пользоваться приложением jupyter notebook для редактирования и запуска кода на python в интерактивном режиме. В этом видео я расскажу вам о том, как устроен наш курс и какие темы мы затронем в ближайшие 7 недель.

Во-первых, каждая неделя будет начинаться с лекционных занятий, на которых будут освещены основные теоретические моменты. Во-вторых, будут практические видео, на которых будут разобраны реализации алгоритмов в различных библиотеках python. Помимо лекций и практических занятий будут тестовые задания. Некоторые из тестовых заданий будут даны для закрепления материала и не будут влиять на итоговую оценку по курсу. Некоторые будут оцениваться и, соответственно, влиять на оценку. Также будут задания на программирование. Эти задания будут выполняться в jupyter notebook. Например, у нас будут задания по работе с матрицами и по обработке обучающих данных, а также на построение тех или иных моделей машинного обучения. Каждое задание на программирование будет сопровождаться подробными инструкциями, что и как нужно делать.

Теперь расскажу об учебном плане курса. На **первой неделе** мы разберемся в основных понятиях в области науки о данных, рассмотрим, какая математика лежит в основе машинного обучения. Мы изучим такие понятия, как векторы, матрицы, функции и производные, которые являются основой для построения алгоритмов машинного обучения.

На последующих неделях я буду рассказывать о признаках, описывающих объекты в наборах данных, о метриках качества для различных алгоритмов, о том, как устроены разные методы машинного обучения, о принципах их работы, различных модификациях и области применения, а также плюсах и минусах.

Вторую неделю мы начнем с обсуждения того, какие шаги включает в себя процесс разработки проекта машинного обучения, начиная от определения задачи и сбора данных и заканчивая развертыванием и обслуживанием модели. Кроме того, вторая неделя будет посвящена различным методам построения признаковых пространств для описания объектов в наборах данных.

Будет рассмотрена тема о метриках качества регрессии в задачах машинного обучения. Также на второй неделе мы начнем изучение класса алгоритмов, называемых обучением с учителем. Обучение с учителем (англ. Supervised learning) - это тип машинного обучения, в котором модель обучается на основе размеченных данных, где каждый пример входных данных сопоставлен с известным, размеченным выходом. Мы поговорим о методе ближайших соседей, одном из самых простых и интуитивно понятных алгоритмов в машинном обучении.

На **третьей неделе** мы разберем такой метод, как линейная регрессия, поговорим о том, что такое переобучение и регуляризация моделей. Кроме этого, мы обсудим такие понятия, как функция потерь и градиентный спуск и как с их помощью численно найти локальный минимум дифференцируемой функции. Градиентный спуск — это основной метод для оптимизации многих алгоритмов в машинном обучении.

Четвертая неделя будет посвящена методам линейной классификации. Здесь мы рассмотрим такие модели, как логистическая регрессия и метод опорных векторов. Также на четвертой неделе будут разобраны метрики качества классификации.

На **пятой неделе** мы поговорим о решающих деревьях. Решающее дерево позволяет делать предсказания целевой переменной, используя для этого последовательность простых решающих правил. Такие правила называют предикатами. Шаги по предсказанию ответа с помощью решающего дерева напоминают естественный для человека процесс принятия решений. В лекциях будет рассказано о том, как строить деревья и подбирать предикаты, а также о композициях моделей на основе решающих деревьев.

Темой **шестой недели** будет градиентный бустинг. Градиентный бустинг является одним из наиболее популярных алгоритмов машинного обучения, используемых на практике, наряду с нейронными сетями. Идея бустинга основана на построении композиции из простых моделей (как правило, деревьев) для получения сильной обучающей модели. Алгоритм бустинга заключается в том, что каждая следующая модель в композиции пытается исправить ошибки, допущенные предыдущими моделями, постепенно улучшая качество предсказания. Мы обсудим особенности градиентного бустинга, его настройку и использование.

На **седьмой неделе** мы поговорим о задачах обучения без учителя. Обучение без учителя (англ. Unsupervised learning) - это набор подходов в машинном обучении, при котором модель пытается найти закономерности в неразмеченных данных. Основные задачи, решаемые с помощью обучения без учителя: кластеризация (группировка объектов на основании их похожести), снижение размерности данных (методы PCA, t-SNE) и другие. Мы познакомимся с основными алгоритмами unsupervised learning и их применением на практике.

Мы будем работать с со следующими библиотеками:

- numpy - библиотека python для матричных операций, статистических функций, функций линейной алгебры
- sympy - библиотека python для работы с символьными выражениями, математическими символами и уравнениями
- scipy - библиотека python для работы с массивами, линейной алгеброй, статистикой, оптимизацией
- matplotlib - библиотека python для для создания графиков и визуализации данных
- pandas - библиотека python для работы с табличными данными
- scikit-learn - библиотека python для реализации базовых алгоритмов машинного обучения

0.2 Основные термины в области искусственного интеллекта и машинного обучения

Прежде чем мы приступим к изучению математики и алгоритмов машинного обучения, давайте рассмотрим основные термины в этой области.

0.2.1 Общие термины

- **Линейная алгебра (англ. Linear algebra):** — это раздел математики, который изучает линейные пространства и линейные отображения между ними. Линейная алгебра — основа для многих методов машинного обучения, таких как линейная регрессия, метод главных компонент (PCA), SVM, нейронные сети и многое другое.
- **Математический анализ (англ. Calculus):** — дисциплина, которая занимается изучением функций и их свойств. Математический анализ включает в себя изучение границ, непрерывности, дифференцируемости и интегрирования функций. В машинном обучении используется для оптимизации функций потерь, обучения моделей и других аспектов.
- **Численные методы (англ. Numerical analysis):** это набор методов для приближенного решения математических задач. Они используются в машинном обучении для решения задач оптимизации, аппроксимации функций и т.д.
- **Теория вероятностей и математическая статистика (англ. Probability theory and mathematical statistics):** — дисциплины, которые позволяют оценивать вероятности событий и делать выводы на основе данных. В машинном обучении они используются для создания моделей, оценки их точности, проверки гипотез и т.д.
- **Искусственный интеллект (ИИ, англ. Artificial intelligence, AI)** — это область информатики, которая занимается разработкой алгоритмов и систем, способных выполнять задачи, которые обычно требуют умственных способностей человека, таких как распознавание речи и изображений, обработка естественного языка, принятие решений на основе данных и многие другие. Термин «искусственный интеллект» включает в себя множество областей, таких как машинное обучение, нейронные сети и глубинное обучение. Они используются для создания алгоритмов, которые способны выполнять сложные задачи, которые ранее могли выполнять только люди. Имеются две трактовки искусственного интеллекта: широкий искусственный интеллект и узкий искусственный интеллект. Они отличаются друг от друга по уровню способности к решению различных задач.

  Узкий искусственный интеллект ограничен в своих способностях и способен решать только определенный набор задач, для которых он был создан. Эти задачи могут включать в себя распознавание речи, обработку естественного языка, игры, автоматизацию задач и т.д. Примеры узкого ИИ включают в себя голосовые помощники Алиса и Siri, которые могут выполнять ограниченный набор задач, таких как открытие приложений, поиск информации в Интернете, установка таймера и т.д.

  В идеале, **широкий ИИ** не имеет ограничений и может решать задачу из любой области. На данный момент в мире нет моделей, полностью соответствующих таким характеристикам. Существуют системы, в которых комплексно решается несколько задач узкого искуственного интеллекта. К примеру, это модели, используемые в автономных автомобилях, роботах-хирургах и системах управления инфраструктурой. Но стоит заметить, что на данный момент появляются большие модели, которые способны решать очень широкий спектр задач, такие как GPT4. В данном курсе будут изучаться базовые модели узкого искусственного интеллекта.

- **Машинное обучение (англ. Machine learning, ML)** — это подраздел искусственного интеллекта, который позволяет компьютерам учиться и делать предсказания на основе анализа больших объемов данных. Машинное обучение использует алгоритмы и статистические модели, чтобы извлекать информацию из данных и делать предсказания или принимать решения. Использование машинного обучения в различных областях, таких как медицина, финансы, промышленность, розничная торговля и другие, позволяет существенно увеличить эффективность и точность принятия решений, а также автоматизировать рутинные задачи.
- **Глубинное обучение (англ. Deep learning)** — это подраздел машинного обучения, который использует нейронные сети с большим количеством слоев для анализа данных и выявления скрытых закономерностей. Основная идея глубинного обучения заключается в том, чтобы создать модель, которая может автоматически извлекать важные признаки из данных, без необходимости ручного определения этих признаков. Для этого глубинные нейронные сети используют многослойную архитектуру, которая позволяет обрабатывать данные на разных уровнях абстракции. В отличие от традиционных методов машинного обучения, где признаки данных должны быть явно определены и предварительно выбраны, глубинное обучение позволяет моделировать данные с большой выразительностью и гибкостью, что делает его особенно полезным для решения сложных задач, таких как распознавание образов, обработка естественного языка, распознавание речи, анализ временных рядов и многих других.
- **Наука о данных (англ. Data science, DS)** — это междисциплинарная область знаний, которая объединяет методы, инструменты и технологии для извлечения знаний и информации из данных. Data science объединяет знания и методы из таких областей, как математика, статистика, информатика, машинное обучение, базы данных и визуализация данных, чтобы анализировать и извлекать знания из больших, разнообразных, сложных и неструктурированных данных.
- **Анализ данных (англ. Data analysis)** - это процесс извлечения полезной информации из данных с целью получения понимания явлений и процессов, лежащих в их основе, а также принятия решений на основе полученных результатов. Анализ данных включает в себя различные методы, такие как статистический анализ, машинное обучение, исследование данных и другие. Целью анализа данных является выявление закономерностей, паттернов и трендов в данных, выделение важных факторов, влияющих на результаты, и прогнозирование будущих событий.
- **Обучение с учителем (англ. Supervised learning)** — тип задач в машинном обучении, при котором модель обучается на размеченных данных, где каждый пример имеет метку класса или значение целевой переменной.
- **Обучение без учителя (англ. Unsupervised learning)** - тип задач в машинном обучении, при котором модель обучается на неразмеченных данных, где нет меток классов или целевой переменной, и цель состоит в том, чтобы выявить структуру данных и паттерны.
- **Обучение с подкреплением (англ. Reinforcement learning, RL)** - тип задач в машинном обучении, при котором модель обучается взаимодействуя с окружающей средой и получая обратную связь в виде награды или штрафа.

0.2.2 Основные специальные термины

В машинном обучении также существует множество специальных терминов, которые описывают различные концепции, методы и алгоритмы. В первую очередь, приведем основные из них.

Основные компоненты, используемые в алгоритмах машинного обучения

- **Объекты (англ. Objects)** — наблюдаемые данные, которые необходимо обработать и проанализировать. Объекты могут представлять собой любые данные, которые можно описать числами или другими количественными характеристиками, такими как звуковые сигналы, изображения, текстовые документы, числовые таблицы и т.д.
  Каждый объект в машинном обучении представлен набором признаков или характеристик, которые описывают его свойства и состояние. Например, для изображения объектом могут быть пиксели, а для текстового документа - слова или символы.
  Объекты в машинном обучении используются для обучения моделей, которые могут классифицировать или регрессировать новые объекты на основе обучающих данных. Кроме того, объекты могут использоваться для оценки качества моделей и проведения исследовательского анализа данных.
- **Признаки (англ. Features)** — измерения или характеристики объекта, которые используются для его описания и анализа. Признаки могут быть числовыми, бинарными или категориальными, в зависимости от типа данных, которые они описывают. Например, если объектом является человек, то признаками могут быть его возраст, рост, вес, пол, образование, занятость и т.д. Для изображения признаками могут быть яркость и цвет каждого пикселя, а для текста - количество вхождений каждого слова. Признаки играют важную роль в машинном обучении, поскольку на их основе строятся модели для решения задач классификации, регрессии, кластеризации и т.д. Выбор и определение правильных признаков является важным шагом в обработке и подготовке данных, поскольку неинформативные или некорректно выбранные признаки могут привести к плохим результатам моделирования.
- **Целевая переменная (англ. Target variable)** — это переменная, которую модель пытается предсказать или прогнозировать на основе других переменных, называемых признаками или характеристиками. Целевая переменная может быть числовой или категориальной, в зависимости от типа задачи, которую решает модель.
  В задачах регрессии, целевая переменная представляет собой непрерывную числовую переменную, которую модель пытается предсказать. В задачах классификации, целевая переменная представляет собой категориальную переменную, которую модель пытается определить для новых данных. Выбор правильной целевой переменной является критически важным в машинном обучении, так как это определяет тип задачи и алгоритмы, которые будут использоваться для обучения модели.
- **Набор данных (англ. Dataset)** — это коллекция данных, которые используются для обучения и тестирования моделей машинного обучения. Набор данных может содержать множество примеров, каждый из которых представляет собой некоторый объект или явление, для которых уже известны значения признаков и целевых переменных, которые нужно предсказать.
  Для построения модели машинного обучения требуется набор данных, который будет использоваться для обучения (training dataset) и для проверки качества модели

(validation dataset или test dataset). Обычно набор данных разбивается на две части: одна часть используется для обучения модели, а другая - для ее проверки. Размеры частей могут варьироваться, но обычно на обучение выделяется от 70% до 90% данных, а на проверку - от 10% до 30%.

Набор данных должен быть предварительно обработан и очищен от выбросов, пропусков и ошибок. Также может потребоваться преобразование данных для улучшения их качества или для адаптации под конкретный алгоритм машинного обучения. Наборы данных могут быть различных типов: текстовые, изображений, звуковые, видео и другие. Они могут быть собраны из различных источников: измерений, экспериментов, интернета и т.д.

- **Модель** — это математическое представление, которое используется для решения конкретной задачи, например, для предсказания значения целевой переменной на основе данных признаков. Модель обучается на наборе данных, который включает в себя объекты, признаки и (в задачах регрессии и классификации) целевую переменную, и после этого может быть использована для предсказания для новых данных.

  Модель представлена в виде алгоритма, который может быть реализован на компьютере. Обучение модели в машинном обучении заключается в том, чтобы настроить ее параметры таким образом, чтобы минимизировать ошибку прогноза на данных. Этот процесс обучения может быть выполнен различными способами, в зависимости от типа задачи и выбранного алгоритма.

  Существует множество различных типов моделей, используемых в машинном обучении, включая метрические модели, линейные модели, деревья решений, ансамблевые модели, нейронные сети и другие. Каждая модель имеет свои сильные и слабые стороны, и выбор конкретной модели зависит от требуемой точности, сложности задачи, объема данных и других факторов.

- **Функция потерь (англ. Loss function)** — математическая функция, которая оценивает разницу между предсказанными значениями модели и фактическими значениями на обучающем наборе данных. Цель функции потерь заключается в том, чтобы минимизировать разницу между предсказанными значениями и фактическими значениями, чтобы модель могла давать наиболее точные прогнозы на новых данных. В задачах регрессии, когда требуется предсказать численное значение (например, цену недвижимости), функция потерь может быть выбрана как среднеквадратическая ошибка (MSE) или средняя абсолютная ошибка (MAE). В задачах классификации, когда требуется предсказать категорию или метку, функция потерь может быть выбрана как перекрестная энтропия (cross-entropy) или логистическая функция потерь.

  В машинном обучении выбор функции потерь зависит от задачи, типа модели и используемого алгоритма обучения. Выбор правильной функции потерь может повлиять на точность модели и скорость ее обучения. Однако, в некоторых случаях может быть необходимо оптимизировать более сложную функцию потерь, которая учитывает не только разницу между предсказанными и фактическими значениями, но и другие факторы, такие как сложность модели или балансировку классов в задаче классификации.

Математическая нотация приведенных выше определений следующая:

- x_i - объект, для которого делается предсказание
- d - размерность
- $\mathbb{X}$ - пространство объектов
- $(x_{i1}, x_{i2}, \ldots, x_{id})$ - набор из d признаков для i-го объекта

- y_i - ответ для объекта i, целевая переменная
- $\mathbb{Y}$ - пространство ответов
- n - количество примеров, размер обучающей выборки
- $X = (x_i, y_i)_{i=1}^n$ - обучающая выборка
- $m(x_i)$ - модель - функция, предсказывающая ответ для объекта x_i. Можно сказать, что m отображает пространство $\mathbb{X}$ в пространство $\mathbb{Y}$
- $\hat{y}_i$ - предсказание модели для объекта i
- $L(y_i, \hat{y}_i)$ - функция потерь - показатель того, насколько предсказание модели отличается от реального ответа на конкретном примере

Основные типы задач в машинном обучении

Давайте разберемся в основных типах задач в машинном обучении.

- **Регрессия (англ. Regression)** — это метод, используемый для анализа отношений между целевой переменной (иногда её называют зависимой переменной) и одной или несколькими независимыми переменными (признаками). Она часто используется для прогнозирования количественных значений, таких как цены, объемы продаж, количество посетителей и т.д.
  В задаче регрессии целевая переменная является количественной, то есть она может принимать любые числовые значения. Цель регрессионного анализа - построить модель, которая может предсказывать значения целевой переменной на основе значений признаков. Для построения модели регрессии используется алгоритм, который находит оптимальные значения параметров модели, минимизирующие ошибку прогнозирования на обучающих данных. Существует множество алгоритмов регрессии, включая метрические методы, линейную регрессию, деревья решений и другие.
  Одной из важных задач в регрессии является оценка качества модели на новых данных, для чего используется тестовый набор данных. Для оценки качества модели применяются различные метрики, такие как среднеквадратичная ошибка (MSE), средняя абсолютная ошибка (MAE) и другие.
- **Классификация (англ. Classification)** — это процесс отнесения объектов к заранее определенным классам или категориям. Классификация используется в широком спектре задач, таких как распознавание образов, определение темы текста, диагностика заболеваний, прогнозирование рынка и многих других.
  В задачах классификации используются алгоритмы машинного обучения, которые обучаются на наборе данных, состоящем из объектов и соответствующих им меток классов (целевых переменных). Обычно для обучения модели используется алгоритм, который находит зависимость между признаками объектов и их метками классов, а затем использует эту зависимость для прогнозирования меток классов для новых объектов.
  Примерами алгоритмов классификации являются логистическая регрессия, метод опорных векторов (SVM), деревья решений, градиентный бустинг и нейронные сети. В зависимости от природы задачи и характеристик данных, один алгоритм может быть более эффективным, чем другой.
  Для оценки качества работы алгоритма классификации используются различные метрики, такие как доля верно предсказанных ответов (accuracy), полнота (recall), точность предсказания (precision), F1-мера и другие. Для выбора наилучшего алгоритма классификации используется оценка метрики на тестовом датасете или кросс-валидация на обучающих данных.

- **Кластеризация (англ. Clusterization)** — это задача разбиения множества объектов на группы (кластеры) таким образом, чтобы объекты внутри каждого кластера были максимально похожи между собой, а объекты из разных кластеров были максимально различными.
  Кластеризация используется во многих областях, таких как маркетинг, медицина, социология, биология и другие. Для выполнения задачи кластеризации используются различные алгоритмы, такие как K-средних (K-means), DBSCAN, иерархическая кластеризация и т.д. Алгоритмы кластеризации могут работать с различными типами данных, включая числовые, категориальные и текстовые.
  Оценка качества кластеризации может быть достаточно сложной задачей, так как нет точного определения того, как должны быть сформированы кластеры. Для оценки качества кластеризации используются различные метрики, такие как коэффициент силуэта, индекс Данна и другие.
- **Снижение размерности (англ. Dimensionality reduction)** — это процесс уменьшения количества признаков в наборе данных, при этом сохраняется максимально возможное количество информации о данных. Это позволяет упростить анализ данных и ускорить вычисления в алгоритмах машинного обучения. Снижение размерности может производиться как для уменьшения количества шума и выбросов в данных, так и для упрощения модели при сохранении информативных признаков. Снижение размерности может быть полезным во многих задачах машинного обучения, таких как классификация, кластеризация, визуализация данных и других.

Помимо приведенных выше типов задач в машинном обучении встречаются и другие (обучение с подкреплением, детектирование аномалий и пр.). Их изучение выходит за рамки тем данного курса.

0.2.3 Вопросы для самопроверки

Математический анализ - это ...:

1. раздел математики, который изучает линейные пространства и линейные отображения между ними
2. раздел математики, который изучает функции и их свойства
3. набор методов численного решения математических задач
4. раздел математики, который изучает вероятности событий

Правильные ответы: 2

Как соотносятся понятия "глубинное обучение "искусственный интеллект"и "машинное обучение"(от большего множества к меньшему):

1. глубинное обучение -> искусственный интеллект -> машинное обучение
2. машинное обучение -> глубинное обучение -> искусственный интеллект
3. искусственный интеллект -> глубинное обучение -> машинное обучение
4. искусственный интеллект -> машинное обучение -> глубинное обучение

Правильные ответы: 4

Для какого тип задач в машинном обучении необходимы метки с правильными ответами для обучения:

1. Обучение с учителем (англ. Supervised learning)
2. Обучение без учителя (англ. Unsupervised learning)
3. Обучение с подкреплением (англ. Reinforcement learning, RL)

Правильные ответы: 1

0.2.4 Резюме по разделу

В этом разделе мы поговорили об основных разделах математики, которые используются в машинном обучении. Мы познакомились с перечнем наиболее часто применяемых в машинном обучении разделов математики: линейная алгебра, математический анализ, оптимизация, численные методы, теория вероятностей, теория графов, теория информации. Математический аппарат позволяет создавать новые методы машинного обучения, представлять методы машинного обучения в виде математических моделей, выражать данные в нужном виде и оптимизировать различные функции.

Мы также рассмотрели основные общие термины в машинном обучении, а именно искусственный интеллект, машинное обучение, глубинное обучение, наука о данных, анализ данных, обучение с учителем, обучение без учителя, обучение с подкреплением. Также мы познакомились с основными специальными терминами, такими как объекты, признаки, целевая переменная, набор данных, модель, функция потерь, регрессия, классификация, кластеризация, снижение размерности.

0.3 Линейная алгебра: векторы

0.3.1 Определение вектора

Вектор — это математический объект, который имеет длину и направление в пространстве. Он может быть использован для представления физических величин, таких как сила, скорость, ускорение и многих других. В машинном обучении векторы являются одним из основных понятий и используются для представления данных. Например, вектор может представлять цвет изображения в RGB-формате, размер объекта на изображении, частоты звука или другие характеристики данных. Векторы также используются в алгоритмах машинного обучения для выполнения операций, таких как классификация, кластеризация и регрессия. Кроме того, векторы могут использоваться для решения задач оптимизации. Каждый объект может быть представлен в виде вектора признаков, где каждый элемент вектора соответствует определенному признаку объекта.

Векторы могут быть обработаны с помощью различных методов машинного обучения, таких как линейная регрессия, метод опорных векторов, нейронные сети, деревья решений и многие другие. При обучении моделей машинного обучения векторы используются для поиска закономерностей, связей и корреляций между признаками, которые могут быть использованы для прогнозирования или классификации новых данных. Векторы используются для создания матрицы признаков, которая является основным входом для большинства алгоритмов машинного обучения. В python распространенный способ представления вектора - это одномерный массив numpy.

Векторы обычно обозначаются либо с помощью жирного шрифта, либо с помощью стрелки над символом: $\mathbf{v}$ или $\vec{v}$. В общем случае, в n-мерном пространстве вектор представляется как набор из n чисел. Компоненты вектора могут быть действительными или комплексными числами. Для записи компонент вектора обычно используют формат столбца (вектор-столбец), где каждый элемент записывается в новой строке:

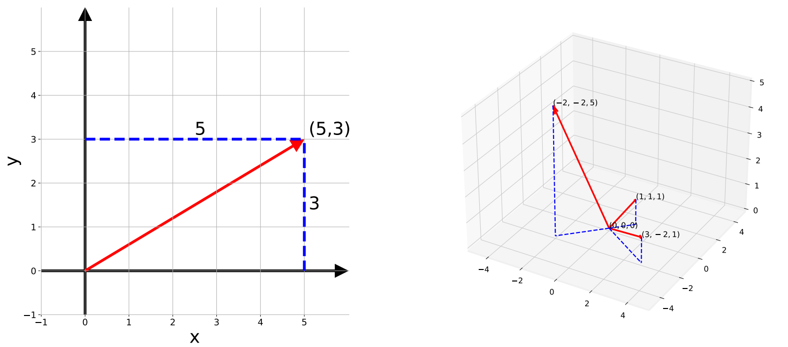
$$\mathbf{v} = \vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

Элементы вектора $v_1,v_2,v_3,\ldots,v_{n-1},v_n$ называются его **координатами** или **компонентами**. Размерностью вектора является количество его компонент.

Существует связь между понятием вектора в n-мерном пространстве и упорядоченной последовательностью действительных чисел $v_1,v_2,v_3,\ldots,v_{n-1},v_n$. В двумерном пространстве, к примеру, вектор может быть представлен как пара чисел (v_1,v_2), где v_1,v_2 могут быть интерпретированы как координаты x, y на плоскости (Рисунок 1 (1)). В трехмерном пространстве векторы задаются с помощью трех координат (Рисунок 1 (2)). Векторы большей размерности не могут быть визуализированы.

Некоторые типы векторов имеют собственные названия. Например, нулевой вектор – это вектор, все компоненты которого равны нулю:

$$\vec{0} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$



(1) (2)

Рисунок 1: (1) — Вектор с координатами (5,3) в двумерном пространстве, (2) — несколько векторов в трехмерном пространстве

В **машинном обучении** принято задавать векторы с помощью только одной точки в заданном векторном пространстве. Соответственно, полагается, что **начало всех векторов находится в точке начала координат**. Такая интерпретация векторов делает интуитивно понятными различные операции над векторами, например, вычисление угла между двумя векторами (Рисунок 2).

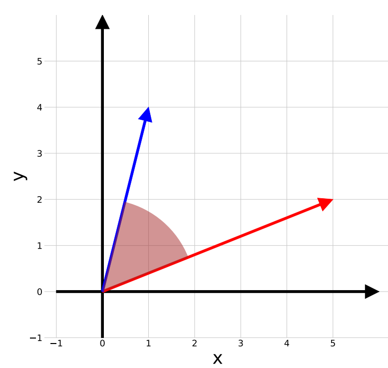


Рисунок 2: Два вектора, исходящих из начала координат

Рассмотрим основные операции над векторами и их геометрический смысл.

0.3.2 Операции над векторами

Определение направления вектора

Направление вектора определяется по направлению от его начала к концу. Другими словами, направление вектора - это угол, под которым он направлен относительно некоторой начальной точки (обычно начало координат) в пространстве. Обычно угол измеряется в градусах или радианах. Например, для вектора с координатами (5,4), угол между горизонтальной осью и вектором может быть найден, используя тангенс угла наклона, который равен $4/5$. Чтобы определить угол по тангенсу, можно воспользоваться обратной функцией тангенса -

арктангенсом (или тангенсом$^{-1}$). Обозначается арктангенс как arctan или tan^{-1}. Если дано значение тангенса угла α, то угол α можно найти следующим образом:

$$\alpha = \arctan(\tan\alpha)$$

Например, если дано, что $\tan\alpha = 4/5$, то угол α можно найти следующим образом:

$$\alpha = \arctan(0.8) \approx 38.659^\circ$$

Однако необходимо помнить, что арктангенс имеет ограничения на область определения и может возвращать только углы в определенном диапазоне значений. Например, для обычной арктангенс функции область определения ограничена значениями $-\frac{\pi}{2} < \arctan(x) < \frac{\pi}{2}$, то есть функция возвращает только углы, лежащие в этом диапазоне. Если необходимо получить угол, лежащий в другом диапазоне, можно использовать дополнительные математические операции, например, добавлять или вычитать π (180 градусов) из значения арктангенса в зависимости от того, в какой четверти находится искомый угол.

Соответственно, направление вектора с координатами (5,4) равно $\approx 38.659^\circ$ (Рисунок 3).

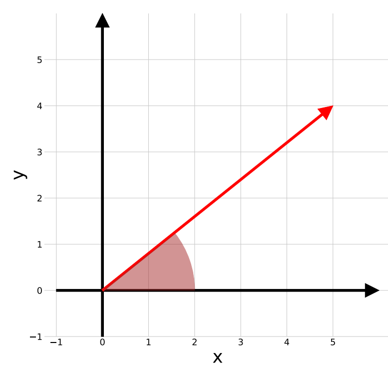


Рисунок 3: Направление вектора, выраженное в градусах

Вычисление длины вектора

Каждый вектор в векторном пространстве, кроме 0-вектора, имеет длину, длина обозначается термином *норма*. Существуют несколько способов определения нормы векторов, но все они моделируют расстояния, которые мы используем в повседневной жизни. Один из способов - это **L1-норма** (Расстояние городских кварталов, Манхэттенская метрика), которая соответствует расстоянию, которое бы проехал автомобиль между начальной и конечной точками по городским улицам, расположенным под углом 90°. Это расстояние находится путем перемещения только вдоль осей координат (Формула 1):

$$||\vec{v}||_1 = (\sum_{i=1}^{d} |v_i|^1)^{\frac{1}{1}}, \tag{1}$$

где v - вектор размерности d

В этой формуле происходит суммирование модулей всех координат. Корень 1-й степени означает отсутствие математических действий над выражением и используется лишь для унификации с метрикой Минковского (Формула 3). К примеру, расстояние от точки (0,0) до точки (5,4) может быть может быть пройдено по разным траекториям, но общее расстояние останется равным 9.

Другой способ - это **L2-норма**, которая соответствует расстоянию, которое пролетел бы вертолет между началом координат и точкой вектора. Это расстояние находится по теореме Пифагора. Сначала происходит суммирование квадратов всех координат, а затем вычисляется корень 2-й степени из этой суммы (Формула 2):

$$||\vec{v}||_2 = (\sum_{i=1}^{d} |v_i|^2)^{\frac{1}{2}}, \tag{2}$$

где v - вектор размерности d

L2-норма является более популярной по сравнению с L1-нормой, потому что она более естественна для измерения длины (Рисунок 4).

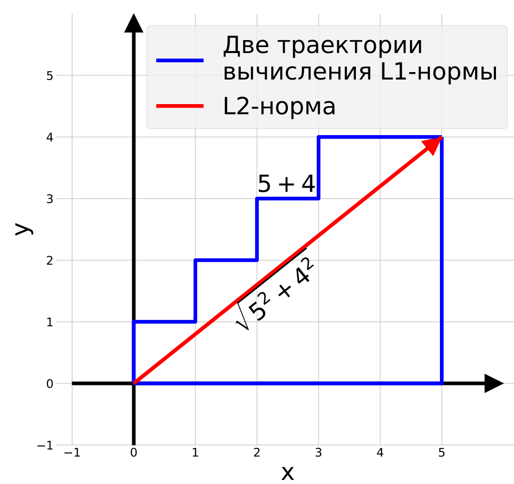


Рисунок 4: Способы вычисления длины вектора

Общая же формула нормы вектора в $\mathbb{R}^d$ носит название метрики Минковского и вычисляется по формуле (Формула 3):

$$||\vec{v}||_n = (\sum_{i=1}^{d} |v_i|^n)^{\frac{1}{n}}, \text{где } v \text{ - вектор размерности } d, \, n \text{ - показатель степени} \tag{3}$$

Поясним, что запись $\mathbb{R}^d$ означает, что мы рассматриваем векторы, элементы которых являются числами из множества действительных чисел $\mathbb{R}$, и каждый вектор имеет d компонент (или размерность d). Таким образом, векторы в $\mathbb{R}^d$ можно записать в виде упорядоченных наборов чисел вида $(x_1, x_2, \ldots, x_d)$, где каждое x_i принадлежит множеству действительных чисел $\mathbb{R}$. Примеры:

- Вектор $(2, 5, -3)$ принадлежит векторному пространству $\mathbb{R}^3$.
- Вектор $(1, 0, 1, -2)$ принадлежит векторному пространству $\mathbb{R}^4$.

Сложение и вычитание векторов

Векторы, так же как и числа, можно складывать:

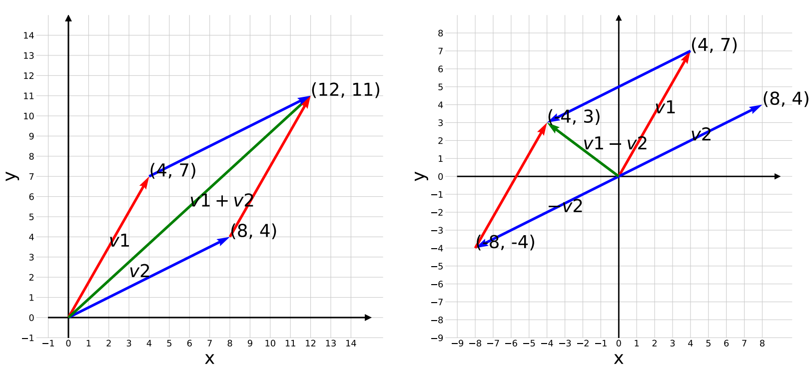
$$\vec{v1}+\vec{v2}=\begin{bmatrix} v1_1 \\ v1_2 \\ \vdots \\ v1_n \end{bmatrix}+\begin{bmatrix} v2_1 \\ v2_2 \\ \vdots \\ v2_n \end{bmatrix}=\begin{bmatrix} v1_1+v2_1 \\ v1_2+v2_2 \\ \vdots \\ v1_n+v2_n \end{bmatrix}$$

и вычитать друг из друга, получая новые векторы:

$$\vec{v1}+(-\vec{v2})=\begin{bmatrix} v1_1 \\ v1_2 \\ \vdots \\ v1_n \end{bmatrix}+\begin{bmatrix} -v2_1 \\ -v2_2 \\ \vdots \\ -v2_n \end{bmatrix}=\begin{bmatrix} v1_1-v2_1 \\ v1_2-v2_2 \\ \vdots \\ v1_n-v2_n \end{bmatrix}$$

Например, для того чтобы сложить вектор $\vec{v1}$ с координатами (4,7) (на рисунке 5 (1) красного цвета) и вектор $\vec{v2}$ с координатами (8,4) (на рисунке 5 (1) синего цвета), нужно просто сложить их соответствующие координаты, и тогда получится новый вектор $v1 \vec{+} v2$ с координатами (12,11) (на рисунке 5 (1) зеленого цвета). Геометрически это можно представить так: суммарный вектор является диагональю параллелограмма, образованного векторами $\vec{v1}$ и $\vec{v2}$ (Рисунок 5 (1)).

Вычитание происходит аналогичным образом, только используется другая диагональ параллелограмма. Например, разность между векторами $\vec{v1}$ (4,7) (на рисунке 5 (2) красного цвета) и $\vec{v2}$ (8,4) (на рисунке 5 (2) синего цвета) вычисляется также поэлементно, получается вектор $v1 \vec{-} v2$ (на рисунке 5 (2) зеленого цвета) с координатами (-4,3) (Рисунок 5 (2)):



(1) (2)

Рисунок 5: (1) — Сложение векторов, (2) — вычитание векторов

Свойства сложения векторов:
Для любых векторов одинакового размера выполнено:

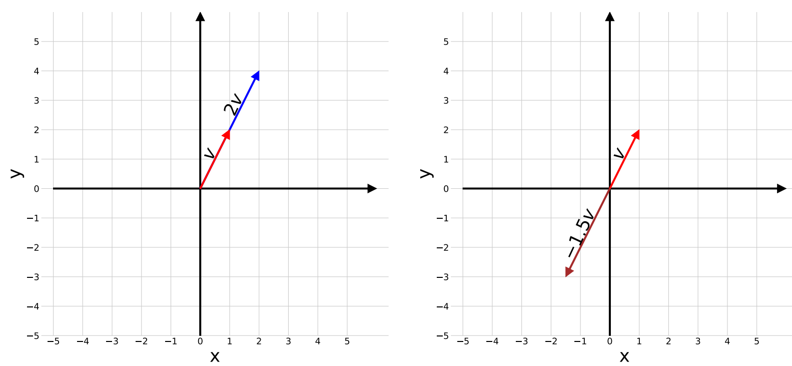
- $\vec{v1}+\vec{v2}=\vec{v2}+\vec{v1}$ (коммутативность сложения)
- $(\vec{v1}+\vec{v2})+\vec{v3}=\vec{v1}+(\vec{v2}+\vec{v3})$ (ассоциативность сложения)

Умножение вектора на константу

Умножение вектора на константу — это поэлементное умножение компонент вектора на данную константу.

$$c\vec{v} = c \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} cv_1 \\ cv_2 \\ \vdots \\ cv_n \end{bmatrix}$$

Умножение вектора на константу приводит к изменению длины вектора и/или его направления, в зависимости от значения константы. Геометрический смысл умножения вектора на константу следующий (Рисунок 6):



(1) (2)

Рисунок 6: (1) — константа положительная, умножение приводит к увеличению длины вектора, направление не меняется, (2) — константа отрицательная, длина вектора также увеличивается, но направление меняется на противоположное

Умножение вектора на число может использоваться для масштабирования векторов в графическом программировании, изменения скорости или ускорения объектов в физических симуляциях, применения линейных преобразований в линейной алгебре, и т.д.

Свойства умножения вектора на константу:
Для любого вектора выполнено:

- $c \cdot \vec{v} = \vec{v} \cdot c$ (коммутативность умножения на константу)
- $(c_1 c_2) \cdot \vec{v} = c_1 \cdot (c_2 \cdot \vec{v})$ (ассоциативность умножения на несколько констант)

Транспонирование вектора

Для вектора применима операция транспонирования. Транспонирование вектора - это операция, при которой вектор записывается в виде матрицы размера 1 x n (если вектор-столбец) или n x 1 (если вектор-строка), где n - количество элементов вектора.

При транспонировании вектор-столбца получается вектор-строка:

$$\vec{v} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$$

Скалярное произведение векторов

Скалярное произведение векторов - это операция, результатом которой является скалярное значение (число), полученное путем умножения соответствующих элементов двух векторов и их последующей суммы.

Скалярное произведение может быть вычислено, только если длины двух векторов равны.

Для скалярного произведения векторов $\vec{v1}$ и $\vec{v2}$ используется одно из следующих обозначений: $\langle\vec{v1},\vec{v2}\rangle, (\vec{v1},\vec{v2}), \mathbf{v1}\cdot\mathbf{v2}, \vec{v1}\cdot\vec{v2}$.

Для двух векторов $\vec{v1}$ и $\vec{v2}$ длиной n скалярное произведение может быть вычислено по формуле 4:

$$\langle\vec{v1},\vec{v2}\rangle = \sum_{i=1}^{n} v1_i v2_i, \tag{4}$$

где $\vec{v1} = [v1_1, v1_2, \ldots, v1_n]$ и $\vec{v2} = [v2_1, v2_2, \ldots, v2_n]$ - два вектора длины n.

Скалярное произведение также может быть записано в виде произведения вектора-строки на вектор-столбец.

Свойства скалярного произведения векторов:

- С помощью скалярного произведения может быть вычислена длина вектора $\vec{v}$ как корень из скалярного произведения вектора на себя: $L_2\text{-norm} = \sqrt{\langle v, v\rangle}$.
- С помощью скалярного произведения может быть вычислен угол между двумя векторами $\vec{v1}$ и $\vec{v2}$: $\cos\theta = \frac{\langle\vec{v1}\cdot\vec{v2}\rangle}{|\vec{v1}||\vec{v2}|}$, где θ - угол между векторами, $|\vec{v1}|$ и $|\vec{v2}|$ - длины векторов $\vec{v1}$ и $\vec{v2}$ соответственно.
- С помощью скалярного произведения может быть вычислена проекция вектора $\vec{v2}$ на вектор $\vec{v1}$: $\text{proj}_{\vec{v}_1}\vec{v}_2 = \frac{\langle\vec{v1}\cdot\vec{v2}\rangle}{|\vec{v}_1|^2}\vec{v}_1$(Рисунок 7):

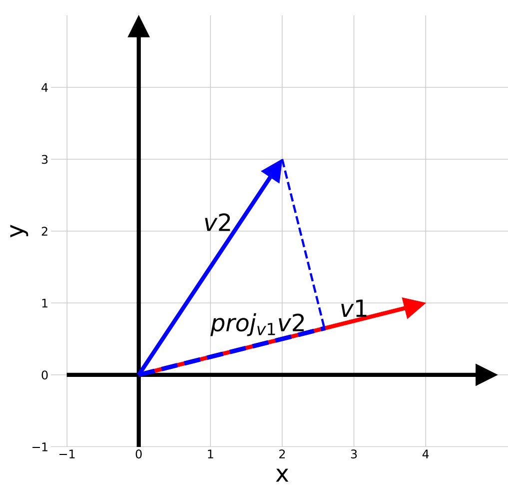


Рисунок 7: Проекция вектора v_2 на вектор v_1

- Для всех векторов, угол между которыми меньше 90°, скалярное произведение будет положительным, для ортогональных векторов (угол между ними равен 90°) скалярное произведение равно 0, для всех векторов, угол между которыми больше 90°, скалярное произведение будет отрицательным (Рисунок 8).

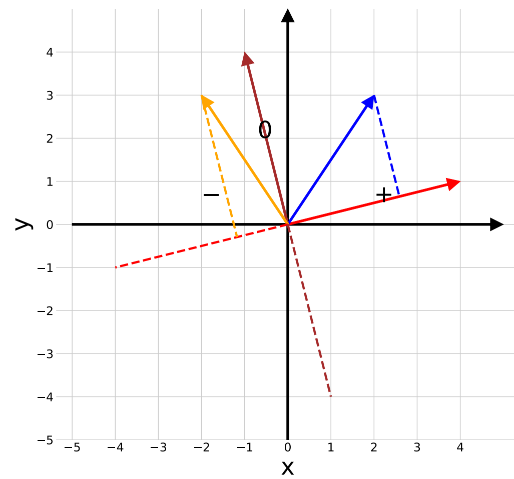


Рисунок 8: Знак скалярного произведения при разных углах между векторами

Скалярное произведение векторов имеет множество применений в математике, физике и инженерии. Некоторые из них:

- Решение систем линейных уравнений: система линейных уравнений может быть записана в виде матричного уравнения $\mathbf{Ax} = \mathbf{b}$, где $\mathbf{A}$ - матрица коэффициентов, $\mathbf{x}$ - вектор неизвестных, $\mathbf{b}$ - вектор правых частей. Решение этой системы может быть получено с помощью методов, основанных на скалярном произведении, например, метод Гаусса.
- Кластерный анализ: в кластерном анализе скалярное произведение может быть использовано для измерения сходства между объектами или для оценки близости кластеров.

Векторное произведение векторов

Векторное произведение - это бинарная операция над двумя векторами, результатом которой является новый вектор, перпендикулярный обоим исходным векторам. На рисунке 9 показано векторное произведение вектора $\vec{v_1}$ с координатами (2, -2, 0) (красный цвет) на вектор $\vec{v_2}$ с координатами (1, 2, 0) (синий цвет). Результатом является вектор $v_1 \vec{\times} v_2$ с координатами (0, 0, 6) (зеленый цвет).

Компоненты вектора, полученного в результате векторного произведения, вычисляются как разности попарных произведений компонент исходных векторов, умноженных на коэффициенты ± 1 в соответствии с правилом буравчика[1]. Векторное произведение двух векторов $\vec{v1}$ и $\vec{v2}$ в трехмерном пространстве вычисляется по формуле 5:

$$\vec{a} \times \vec{b} = \left(a_1 a_2 a_3\right) \times \left(b_1 b_2 b_3\right) = \left(a_2 b_3 - a_3 b_2 a_3 b_1 - a_1 b_3 a_1 b_2 - a_2 b_1\right), \quad \text{где } \vec{a} = (a_1, a_2, a_3) \text{ и } \vec{b} = (b_1, b_2, b_3) \text{ - два вектора в } \mathbb{R}^3. \tag{5}$$

[1] Правило буравчика (винта) для векторного произведения: «Если отобразить векторы так, чтобы их начала совпадали и вращать первый вектор-сомножитель кратчайшим образом ко второму вектору-сомножителю, то буравчик (винт), вращающийся таким же образом, будет завинчиваться в направлении вектора-произведения».

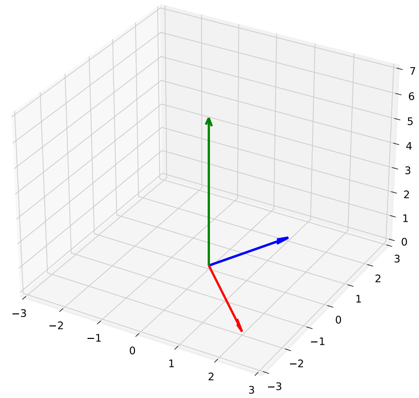


Рисунок 9: Векторное произведение векторов

Векторное произведение используется в машинном обучении довольно редко, но все же может быть полезным для некоторых задач, например, для создания новых признаков из двух или более существующих признаков.

Вычисление расстояния между векторами

Определение расстояния между двумя векторами имеет важное применение в машинном обучении. Для функции измерения расстояния между двумя векторами используется термин метрика. Если мы используем метрику для измерения расстояния между векторами, то она должна удовлетворять следующим требованиям:

- Неотрицательность: Метрика должна быть неотрицательной, то есть расстояние между любыми двумя векторами должно быть не меньше нуля. Формально: $d(x,y) \geq 0$ для любых векторов x и y, и $d(x,y) = 0$ тогда и только тогда, когда $x = y$.
- Симметричность: Метрика должна быть симметричной, то есть расстояние между векторами x и y должно быть таким же, как расстояние между y и x. Формально: $d(x,y) = d(y,x)$ для любых векторов x и y.
- Неравенство треугольника: Метрика должна удовлетворять неравенству треугольника, то есть расстояние между любыми двумя векторами, через третий вектор, должно быть меньше, чем сумма расстояний между первым и третьим векторами, и между вторым и третьим векторами. Формально: $d(x,y) \leq d(x,z) + d(z,y)$ для любых векторов x, y и z.
- Тождественность: Метрика должна удовлетворять тождеству: $d(x,y) = 0$ тогда и только тогда, когда $x = y$.

Метрика, удовлетворяющая этим требованиям, называется метрикой расстояния.

Мы можем использовать способы вычисления расстояния, уже рассмотренные для определения длины вектора. В случае двух векторов L1 расстояние вычисляется как расстояние между концами векторов. Если говорить строго, то такое расстояние называется суммой абсолютных значений разности компонент. Оно вычисляется по формуле 6:

$$|\vec{v1} - \vec{v2}|_1 = \sum_{i=1}^{n} |v1_i - v2_i|,$$
$$\text{где } \vec{v1} \text{ и } \vec{v2} \text{ - два вектора размерности } n,$$
$$v1_i \text{ и } v2_i \text{ - их соответствующие компоненты.} \tag{6}$$

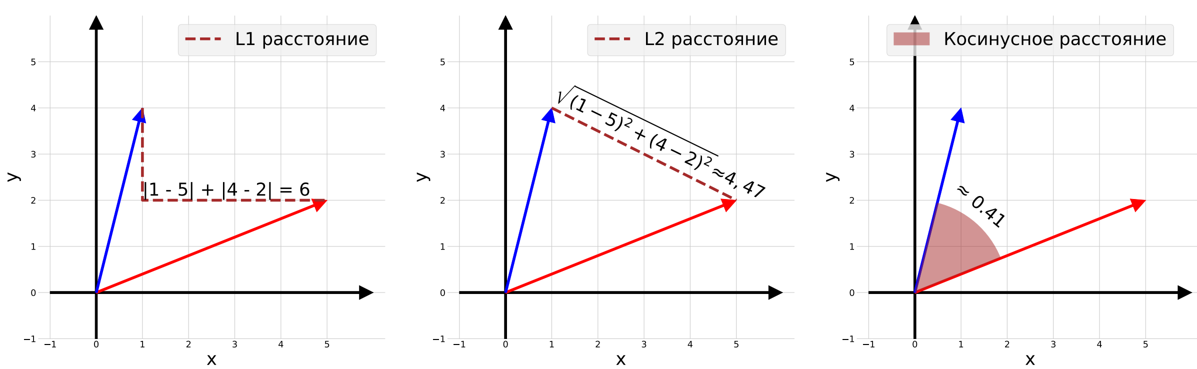
Другой способ - L2-метрика или L2-расстояние - вычисляется как корень квадратный из суммы квадратов разностей компонент по формуле 7:

$$|\vec{v1} - \vec{v2}|_2 = \sqrt{\sum_{i=1}^{n} (v1_i - v2_i)^2},$$
$$\text{где } \vec{v1} \text{ и } \vec{v2} \text{ - два вектора размерности n},$$
$$v1_i \text{ и } v2_i \text{ - их соответствующие компоненты.} \tag{7}$$

Косинусная метрика или косинусное расстояние - это еще один способ определения различий между векторами, вычисляется как 1 - косинус угла между ними по формуле 8:

$$\text{cosine distance}(\vec{v1}, \vec{v2}) = 1 - (\vec{v1} \cdot \vec{v2})/(|\vec{v1}| * |\vec{v2}|),$$
$$\text{где } \vec{v1} \cdot \vec{v2} \text{ - скалярное произведение векторов } \vec{v1} \text{ и } \vec{v2},$$
$$|\vec{v1}| \text{ и } |\vec{v2}| \text{ - длины векторов } \vec{v1} \text{ и } \vec{v2} \tag{8}$$

В машинном обучении знание этих различных мер расстояния между векторами очень полезно для определения сходства между точками данных. Визуальное представления различных мер расстояния показано на рисунке 10:



(1) (2) (3)

Рисунок 10: (1) — L1-расстояние, (2) — L2-расстояние, (3) — косинусное расстояние

L1-расстояние может быть предпочтительнее в задачах, где важна максимальная разница между соответствующими элементами векторов, а L2-расстояние может быть предпочтительнее в задачах, где важна общая длина векторов. Косинусное расстояние часто используется в задачах классификации текстов.

0.3.3 Реализация в Python

Реализация основных операций с векторами рассмотрена в jupyter notebook файле: lecture_01_code_labs_01_vectors.ipynb

0.3.4 Вопросы для самопроверки

Какое из определений вектора НЕ ЯВЛЯЕТСЯ корректным?

1. Вектор - это элемент линейного пространства, который может быть описан с помощью набора координат.
2. Вектор - это упорядоченный набор чисел, который может быть использован для представления физических величин, таких как направление, скорость, ускорение и т.д.
3. Вектор - это математическая величина, которая измеряет скорость изменения функции по отношению к ее аргументу.
4. Вектор - это объект, который используется для описания направления и длины в пространстве.

Правильные ответы: 3

Выберите правильную формулу для вычисления скалярного произведения векторов v1 и v2 в трехмерном пространстве:

1. a b = a1 + b1 + a2 + b2 + a3 + b3
2. a b = a1 * b1 - a2 * b2 - a3 * b3
3. a b = a1 * b1 + a2 * b2 + a3 * b3
4. a b = a1 + a2 + a3 - b1 - b2 - b3

Правильные ответы: 3

Выберите правильный ответ для вычисления L2-нормы между векторами v1 (1,2,2) и v2 (1,5,6) в трехмерном пространстве:

1. L2-норма = 3
2. L2-норма = 4
3. L2-норма = 5
4. L2-норма = 6

Правильные ответы: 3

0.3.5 Резюме по разделу

В этом разделе мы поговорили о том, что такое векторы.

Вектор в математике - это объект, который имеет определенную длину и направление в пространстве.

Вектор может быть представлен в виде упорядоченного набора чисел, называемых компонентами вектора.

Свойства векторов включают длину, направление, сумму и разность векторов, скалярное произведение, векторное произведение и угол между векторами.

0.4 Линейная алгебра: матрицы

0.4.1 Определение матрицы

Матрица - это прямоугольная таблица чисел, символов или выражений, разделенных на строки и столбцы. В машинном обучении термины «таблица» и «матрица» могут использоваться взаимозаменяемо, поскольку матрица - это таблица чисел, упорядоченных в строках и столбцах. В матрицах данные часто представляются в виде числовых значений, которые могут использоваться для обучения алгоритмов машинного обучения. Пример матрицы приведен ниже (Таблица 1):

	Размер	**Цвет**	**Уши**	**Бивень**
Слон	Очень большой	Серый	Круглые	Два
Носорог	Большой	Серый	Овальные	Один
Кот	Маленький	Любой	Треугольные	Нет

Таблица 1: Пример: матрица A

Матрица может быть представлена как набор векторов-строк или векторов-столбцов, каждый из которых содержит элементы матрицы.

Матрица обычно обозначается заглавными буквами латинского алфавита, например, A, B или C. Размеры матрицы называются её размерностью и указывается в виде m×n, где m - количество строк, а n - количество столбцов. Формальная запись для матрицы так: $A^{m\times n} = (a_{ij}) \in \mathbb{R}^{m\times n}$. Например, размерность матрицы 1 равна $A^{3\times 4}$ (3 объекта × 4 признака).

Каждый элемент матрицы может быть обозначен индексами $(i{,}j)$, где i - номер строки, а j - номер столбца. Например, элемент матрицы $A_{2,4}$ равен «Один».

Обычно для обозначения матриц используются квадратные скобки, но возможно также другие скобки и вертикальные линии:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} = \begin{Bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{Bmatrix} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} = \begin{Vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{Vmatrix}$$

Матрицы в машинном обучении используются для представления наборов данных, где каждая строка представляет отдельный пример или объект, а каждый столбец - отдельный признак или характеристику, а также для обучения моделей. Матрицы могут быть обработаны и использованы для обучения различных моделей машинного обучения, таких как линейная регрессия, деревья решений, метод опорных векторов, нейронные сети и многие другие (Рисунок 11). При обучении моделей машинного обучения матрицы используются для поиска закономерностей и корреляций между признаками, которые могут быть использованы для прогнозирования или классификации новых данных. В машинном обучении используются различные операции над матрицами, такие как умножение матриц, транспонирование, инверсия, сингулярное разложение, вычисление ковариационной матрицы и другие. Матрицы также используются для представления различных преобразований данных, таких как нормализация, стандартизация и сжатие данных.

На рисунке 11 показан пример использования матриц в машинном обучении. Обрабатываемое изображение превращается в матрицу, если изображение одноканальное, или тензор (многомерную матрицу), если в изображении более одного цветового канала. Далее из изображения извлекаются признаки (также с помощью матриц), после чего модель может делать предсказания.

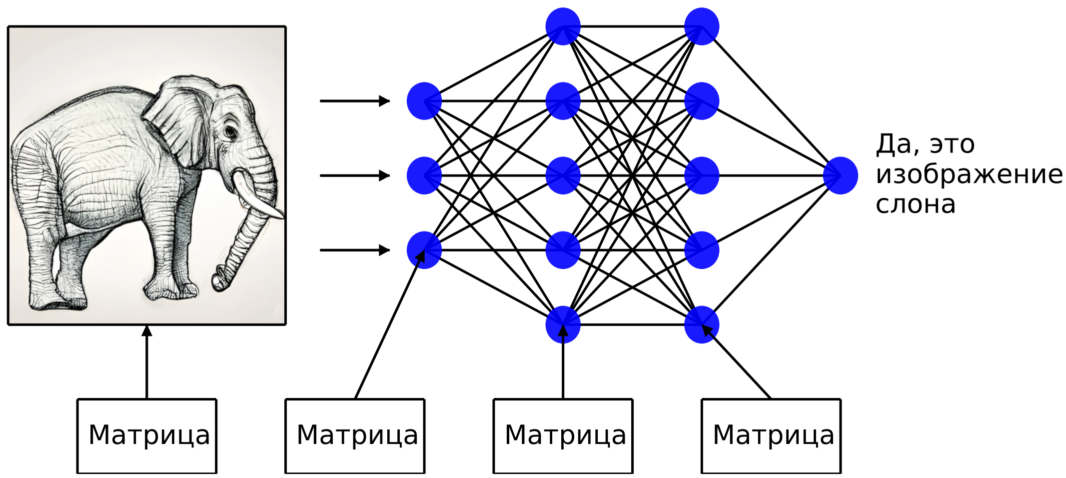


Рисунок 11: Матрицы — основа алгоритмов ИИ

0.4.2 Операции над матрицами

Сложение и вычитание матриц

Чтобы сложить или вычесть две матрицы A и B, они должны иметь одинаковое количество строк и столбцов. Если бы мы имели дело с таблицами, мы бы сказали, что все ячейки попарно складываются друг с другом. Если говорить строго, то к элементу матрицы A прибавляется или вычитается соответствующий элемент матрицы B:

$$A \pm B = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \pm \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ b_{21} & \cdots & b_{2n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{bmatrix} = \begin{bmatrix} a_{11} \pm b_{11} & \cdots & a_{1n} \pm b_{1n} \\ a_{21} \pm b_{21} & \cdots & a_{2n} \pm b_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} \pm b_{m1} & \cdots & a_{mn} \pm b_{mn} \end{bmatrix}$$

Умножение матрицы на константу

Чтобы выполнить умножение матрицы A на число c, нужно умножить каждый элемент матрицы A на число c. Результатом будет новая матрица.

$$c \cdot A = c \cdot \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} c \cdot a_{11} & c \cdot a_{12} & \cdots & c \cdot a_{1n} \\ c \cdot a_{21} & c \cdot a_{22} & \cdots & c \cdot a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c \cdot a_{m1} & c \cdot a_{m2} & \cdots & c \cdot a_{mn} \end{bmatrix}$$

Транспонирование матрицы

Транспонирование матрицы - это операция, при которой строки матрицы становятся ее столбцами, а столбцы - строками, то есть матрица отражается относительно ее главной диагонали[1]. Обозначение для транспонированной матрицы - символ T в верхнем правом углу:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad A^{\intercal} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Произведение матриц

Умножение матрицы на матрицу - это одна из основных операций линейной алгебры. Умножать матрицы можно только в том случае, когда вторая размерность первой матрицы равна первой размерности второй матрицы. Для того, чтобы умножить матрицу A размерности m×n на матрицу B размерности n×p, необходимо выполнить следующие действия:

1. Создать новую матрицу C, которая будет содержать результат умножения матриц A и B. Новая матрица должна иметь размерность m×p (первая размерность первой матрицы × вторая размерность второй матрицы)
2. Для каждого элемента матрицы $C_{i,j}$ вычислить его значение как сумму произведений элементов i-й строки матрицы A на соответствующие элементы j-го столбца матрицы B: $C_{i,j} = A_{i,1} * B_{1,j} + A_{i,2} * B_{2,j} + \cdots + A_{i,n} * B_{n,j}$

Для того, чтобы запомнить, как происходит умножение матрицы на матрицу, можно воспользоваться алгоритмом, изображенным на рисунке 12. Первую матрицу (матрица A) необходимо поместить слева от итоговой (матрица C), вторую (матрица B) — сверху.

Использование единичной и обратной матриц

Единичная матрица - это квадратная матрица, у которой на главной диагонали стоят единицы, а остальные элементы равны нулю. Обычно единичную матрицу обозначают буквой I:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{или} \qquad I_{3\times 3}$$

Обратная матрица - это матрица, умножение которой на исходную матрицу дает единичную матрицу. Обратную матрицу можно обозначить с помощью символа -1 в верхнем правом углу матрицы. Пример исходной, обратной и единичной матриц представлени ниже (Уравнение 9):

$$A = \begin{bmatrix} 2 & 5 & 7 \\ 6 & 3 & 4 \\ 5 & -2 & -3 \end{bmatrix} \quad A^{-1} = \begin{bmatrix} 1 & -1 & 1 \\ -38 & 41 & -34 \\ 27 & -29 & 24 \end{bmatrix} \quad I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

[1] Главная диагональ матрицы — это последовательность элементов матрицы, которые расположены на диагонали, идущей от верхнего левого угла матрицы. Главная диагональ состоит из элементов, расположенных на позициях (1,1), (2,2), (3,3) и т.д.

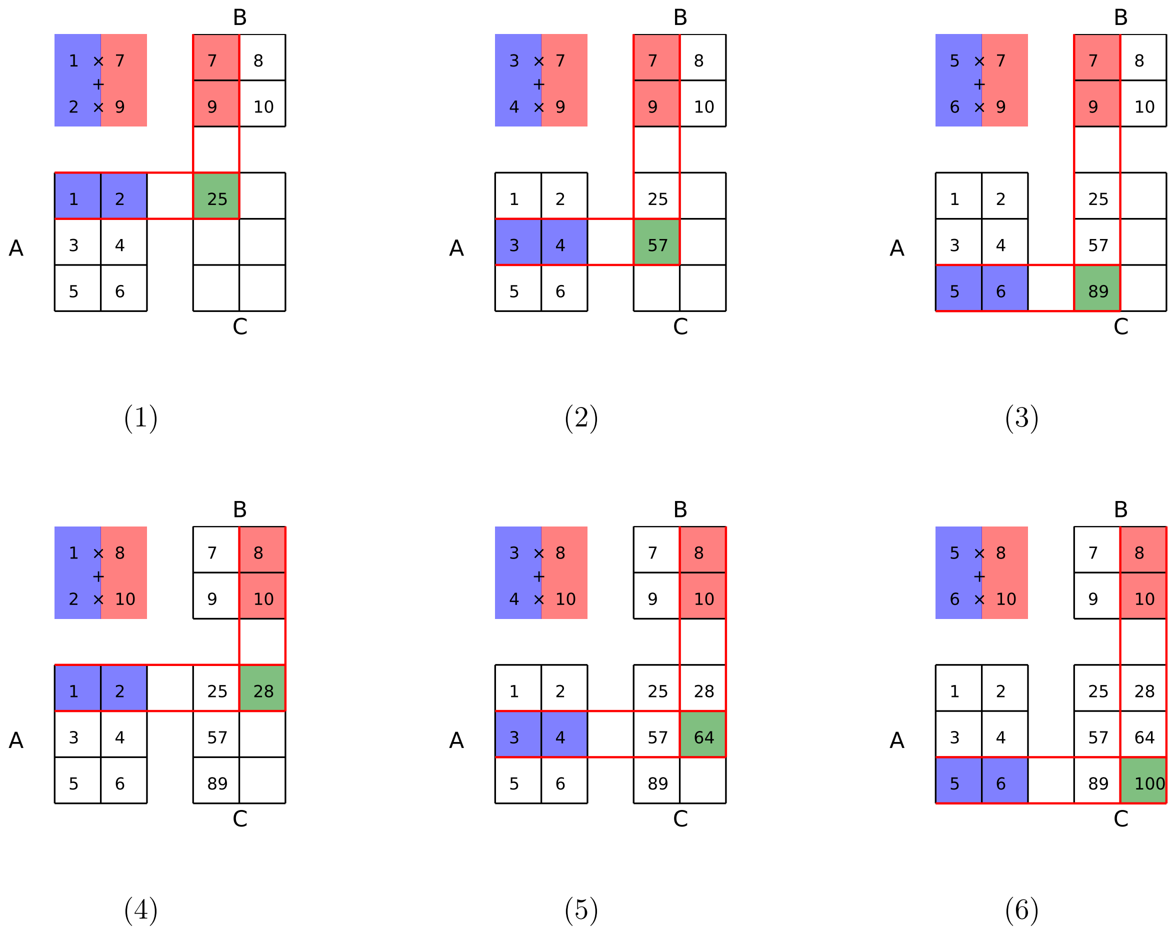


Рисунок 12: Визуализациа процесса умножения матрицы на матрицу

Единичная и обратная матрицы играют важную роль в линейной алгебре, которая является фундаментальной для многих методов машинного обучения. Единичная матрица используется, например, при решении систем линейных уравнений. Обратная матрица также используется при решении задач оптимизации и настройки параметров моделей машинного обучения. Однако, стоит отметить, что обратная матрица не всегда может быть вычислена. Например, если определитель (мы обсудим, что это, ниже) матрицы равен нулю, то обратной матрицы не существует. Это может возникнуть, например, когда количество признаков превышает количество наблюдений в выборке. Поэтому важно тщательно проверять матрицы на обратимость и использовать методы, которые не требуют обращения матриц, если это возможно.

Выявление мультиколлинеарности в данных

Мультиколлинеарность - это явление в статистике и анализе данных, когда **две или более переменных в модели сильно коррелируют друг с другом.**

Простейший пример: в наборе данных присутствуют два признака, один из которых описывает расстояние в метрах, другой - в километрах (Таблица 2):

	Расстояние до Москвы, км	**Расстояние до Москвы, м**	**Население**	**Площадь, км^2**
Воронеж	466	466 000	1 057 000	596
Саратов	725	725 000	901 000	393
Тамбов	419	419 000	290 000	107

Таблица 2: Пример мультиколлинеарности в данных: расстояние в метрах и километрах линейно зависимы

Эти признаки линейно связаны друг с другом, так что невозможно однозначно определить влияние каждой переменной на целевую переменную. Матрица, содержащая линейно зависимые строки, называется сингулярной или вырожденной. Матрица, не содержащая линейно зависимых строк, называется невырожденной или обратимой.

Линейная зависимость признаков в машинном обучении порождает несколько проблем:

- Переопределение модели: если признаки линейно зависимы, то можно построить множество моделей, которые дадут одинаковый результат. Это может привести к тому, что модель будет слишком сложной и переобученной, т.е. она будет хорошо работать на обучающих данных, но плохо на новых данных.
- Низкая устойчивость: если признаки линейно зависимы, то модель может быть чувствительна к небольшим изменениям входных данных. Например, если один признак можно выразить через другой с небольшой ошибкой, то небольшое изменение входных данных может привести к значительным изменениям в результатах.
- Плохая интерпретируемость: если признаки линейно зависимы, то сложно понять, какой признак вносит больший вклад в результаты модели. Это может быть проблематично для анализа результатов и принятия решений на основе модели.

Мультиколлинеарность может привести к неправильной оценке коэффициентов модели и снижению точности прогнозирования. Кроме того, она может сделать модель менее интерпретируемой, что затрудняет понимание вклада каждой переменной в модель. Если мультиколлинеарность обнаруживается, можно принять меры, такие как исключение одной из коррелирующих переменных из модели, объединение коррелирующих переменных в одну переменную или использование регуляризации для уменьшения влияния некоторых переменных на модель.

Для выявления мультиколлинеарности можно использовать статистические методы, такие как корреляционный анализ или метод главных компонент. Далее мы рассмотрим такие методы, как вычисление **ранга** и **определителя** матрицы для выявления мультиколлинеарности в данных. Если матрица признаков имеет низкий ранг или близкий к нулю определитель, это может указывать на наличие мультиколлинеарности в данных.

Ранг матрицы

Ранг матрицы - это количество линейно независимых строк или столбцов в матрице. Если матрица имеет ранг, равный меньшему из числа строк или столбцов, то она называется вырожденной.

Высокий ранг матрицы признаков означает, что в данных мало корреляции между признаками и это может быть признаком отсутствия мультиколлинеарности.

Одной из интерпретаций ранга матрицы является **мера неизбыточной информации**, которая хранится в матрице. Если строка/столбец выражается как линейная комбинация других строк/столбцов, то информацию в строке/столбце можно считать избыточной, соответственно, ранг матрицы снижается на 1.

Рассмотрим это на простом наглядном примере. Допустим, у нас есть три матрицы (Рисунок 13):

Рисунок 13: Вычисление определителя для матрицы 2×2

Информация в первой матрице:

- строка 1: банан желтый
- строка 2: апельсин оранжевый

Информация во второй матрице:

- строка 1: банан желтый
- строка 2: банан желтый

Информация в третьей матрице:

- строка 1: банан
- строка 2: бана

Мы видим, что в первой матрице содержится информация о двух фактах: цвет банана и апельсина. Соответственно, ранг такой системы будет 2. Ранг второй системы будет равен 1. Третья матрица не хранит никакой информации о цвете фруктов, соответственно, её ранг равен 0.

Рассмотрим другой пример, с числовыми значениями(Рисунок 14):

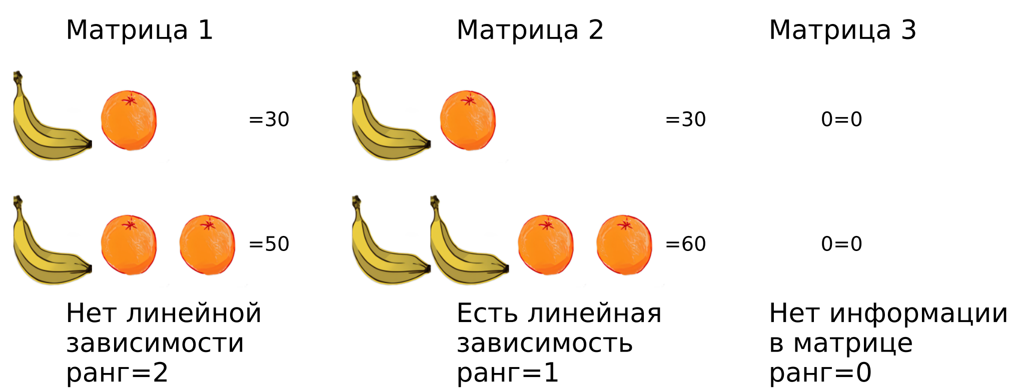


Рисунок 14: Вычисление определителя для матрицы 2×2

- строка 1: 1 банан и 1 апельсин стоят 30 рублей
- строка 2: 1 банан и 2 апельсина стоят 50 рублей

Информация во второй матрице:

- строка 1: 1 банан и 1 апельсин стоят 30 рублей
- строка 2: 2 банан и 2 апельсина стоят 60 рублей

Информация в третьей матрице:

- строка 1: 0 бананов и 0 апельсинов стоят 0 рублей
- строка 2: бананов и 0 апельсинов стоят 0 рублей

Если рассматривать матрицы как системы линейных уравнений, мы видим что в первой матрице нет линейной зависимости строк, и имеется единственное решение: 1 банан стоит 10 рублей и 1 апельсин стоит 20 рублей. Ранг матрицы равен 2. Вторая матрица содержит линейно зависимые строки (строка 2 равна строке 1×2), соответственно, решение системы уравнений сводится к линии. Ранг матрицы равен 1. Третья матрица не содержит никакой информации, её решение сводится к плоскости. Ранг матрицы равен 0.

Существует несколько методов вычисления ранга матрицы:

- Метод Гаусса-Жордана: матрица приводится к ступенчатому виду, затем вычисляется количество ненулевых строк, которое и является рангом матрицы
- Метод элементарных преобразований: матрица приводится к эквивалентной матрице, которая имеет минимальный набор строк и столбцов, образующих базис в пространстве строк (или столбцов) матрицы. Количество таких строк (или столбцов) и будет равно рангу матрицы
- Метод определителей: ранг матрицы равен максимальному порядку ненулевых миноров матрицы
- Метод сингулярного разложения: ранг матрицы равен количеству ненулевых сингулярных значений матрицы
- Метод Грама-Шмидта: матрица приводится к ортонормированной матрице, затем вычисляется количество ненулевых строк, которое и является рангом матрицы

Мы не будем разбирать их подробно (это делается в полноценных курсах по линейной алгебре). Для вычисления ранга матриц мы будем пользоваться библиотекой numpy.

Определитель матрицы

Определитель матрицы (его еще называют детерминантом) - это число, которое вычисляется из элементов **квадратной** матрицы и используется для определения, является ли матрица вырожденной или нет.

Определитель, близкий к нулю, указывает на наличие мультиколлинеарности в данных.

Рассмотрим понятие определителя на двух простых наглядных примерах с матрицами размерностью 2×2 и 3×3.

Допустим у нас есть матрица A:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Определитель будет вычисляться по формуле: a×d - b×c (произведение элементов на главной диагонали матрицы минус произведение элементов на побочной диагонали[1] матрицы). Если он равен 0, то строки линейно зависимы.

Давайте рассмотрим, почему это так (Рисунок 15). Если мы допускаем, что строки линейно зависимы, то строка 2 равна строке 1, умноженной на какой-то коэффициент k. Тогда:

$$a \times k = c, \quad b \times k = d \Rightarrow$$
$$k = \frac{c}{a} = \frac{d}{b} \Rightarrow$$
$$a \times d = b \times c \Rightarrow$$
$$a \times d - b \times c = 0$$

Определитель матрицы 3×3 можно вычислить по формуле Саррюса. Пусть дана матрица:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Тогда определитель можно вычислить следующим образом:

[1] Побочная диагональ матрицы — это последовательность элементов матрицы, которые расположены на диагонали, идущей от верхнего правого угла матрицы. Побочная диагональ состоит из элементов, расположенных на позициях (1,n), (2,n-1), (3,n-2) и т.д., где n - размерность матрицы.

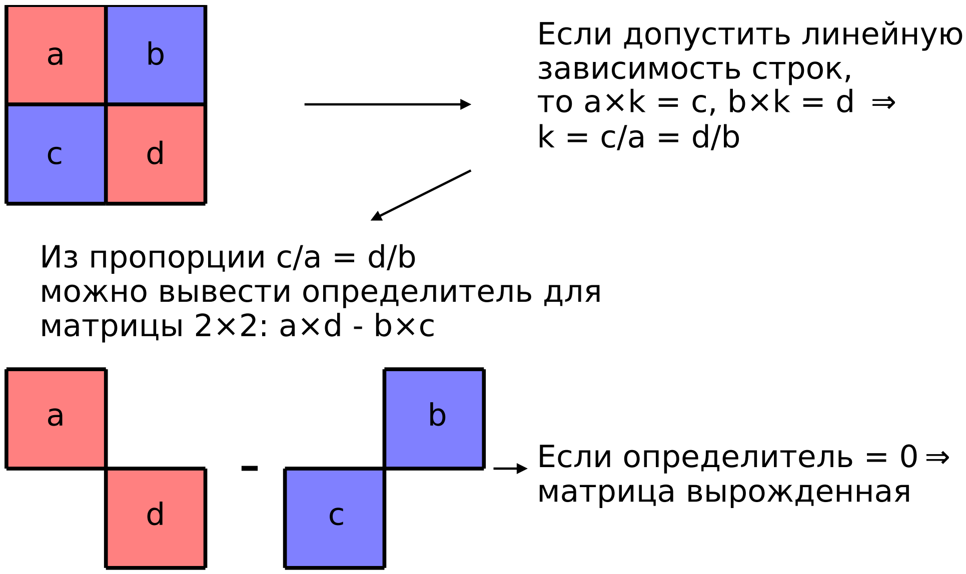


Рисунок 15: Вычисление определителя для матрицы 2×2

1. Создать два вспомогательных столбца путем копирования первых двух столбцов матрицы за третьим столбцом
2. Перемножить элементы по диагоналям, идущим слева направо, и сложить полученные произведения: $a_{11} * a_{22} * a_{33} + a_{12} * a_{23} * a_{31} + a_{13} * a_{21} * a_{32}$
3. Перемножить элементы по диагоналям, идущим справа налево, и вычесть полученные произведения: $-a_{13} * a_{22} * a_{31} - a_{11} * a_{23} * a_{32} - a_{12} * a_{21} * a_{33}$

Результатом будет определитель матрицы: $\det = a_{11} * a_{22} * a_{33} + a_{12} * a_{23} * a_{31} + a_{13} * a_{21} * a_{32} - a_{13} * a_{22} * a_{31} - a_{11} * a_{23} * a_{32} - a_{12} * a_{21} * a_{33}$

Иллюстрация формулы Саррюса приведена на рисунке 16.

0.4.3 Реализация в Python

Реализация основных операций с матрицами рассмотрена в jupyter notebook файле: lecture_01_code_labs_02_matrices.ipynb

На этом мы заканчиваем знакомство с основными инструментами линейной алгебры, применяемыми в машинном обучении. Давайте перейдем к математическому анализу.

0.4.4 Вопросы для самопроверки

Какие из определений матрицы являются корректным?

1. Матрица - это прямоугольная таблица чисел, разбитая на строки и столбцы.

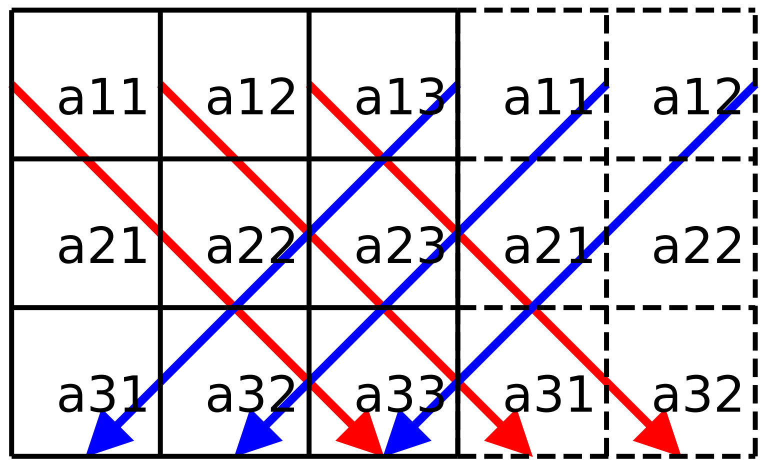


Рисунок 16: Иллюстрация формулы Саррюса

2. Матрица - это математический объект, который используется в машинном обучении для представления наборов данных.
3. Матрица - это математический объект, который используется в машинном обучении для сохранения признаков объектов.
4. Матрица - это последовательность шагов алгоритма машинного обучения.

Правильные ответы: 1,2,3

Транспонирование матрицы - это ...:

1. операция, при которой строки матрицы становятся ее столбцами, а столбцы - строками, матрица отражается относительно ее главной диагонали
2. операция, при которой матрица отражается зеркально по горизонтальной оси
3. операция, при которой матрица отражается зеркально по вертикальной оси
4. операция, при которой в матрицу добавляются новые строки/столбцы

Правильные ответы: 1

Произведение матриц A @ B с какими размерностями допустимо:

1. размерность A = 3x3, размерность B = 5x3
2. размерность A = 5x3, размерность B = 3x5
3. размерность A = 5x5, размерность B = 3x3
4. размерность A = 5x3, размерность B = 5x3

Правильные ответы: 2

0.4.5 Резюме по разделу

В этом разделе мы рассмотрели матрицы. Матрица в машинном обучении - это прямоугольный массив чисел, который представляет собой таблицу, состоящую из строк и столбцов. Некоторые из основных свойств матриц:

- Размерность: количество строк и столбцов, которые определяют размерность матрицы.
- Операции: матрицы можно складывать, вычитать, умножать на число и на другие матрицы.
- Транспонирование: матрица может быть транспонирована, что означает замену строк на столбцы и наоборот.
- Обратная матрица: некоторые матрицы имеют обратную матрицу, которая позволяет решить систему линейных уравнений.
- Вырожденная матрица - это квадратная матрица, у которой определитель равен нулю.
- Определитель матрицы - это число, которое вычисляется из элементов квадратной матрицы и используется для решения линейных систем уравнений, нахождения обратной матрицы, определения линейной зависимости векторов и других задач линейной алгебры.
- Ранг матрицы: максимальное число линейно независимых строк или столбцов в матрице. Ранг матрицы и ее вырожденность тесно связаны. Если матрица вырождена, то ее определитель равен нулю. Следовательно, ранг вырожденной матрицы будет меньше, чем ее размерность.

  Матрицы широко используются в машинном обучении, особенно в задачах обработки данных. Например, в задаче классификации изображений матрица пикселей может быть использована для обучения модели.

0.5 Математический анализ: функции

0.5.1 Определение функции

В математике функция - это соответствие между двумя множествами, где каждому элементу из одного множества (называемого «область определения» (англ. Domain of a function)) сопоставляется единственный элемент из другого множества (называемого «область значений» (англ. Range of a function)).

Область определения функции — это множество всех значений аргумента (переменной x).

Область значений функции - это множество всех значений функции (переменной y). Функция обозначается обычно символом f и записывается в виде f(x), где x - элемент из области определения функции. Область определений и область значений функции могут быть любыми множествами, в том числе и множествами действительных чисел, комплексных чисел, векторов и т.д. Обычно, функции в математическом анализе отображают элементы из множества вещественных чисел на другие вещественные числа. Функции используются для описания различных зависимостей и отношений между различными величинами, например, для определения зависимости скорости от времени, температуры от высоты, или для расчета значения функции в определенной точке.

Функции могут быть заданы различными способами, например, аналитически, графически или таблично.

Аналитически заданная функция представляет собой формулу, которая позволяет вычислить значение функции для любого заданного значения аргумента.

Графически заданная функция представляет собой кривую, которая отображает значения функции на оси координат.

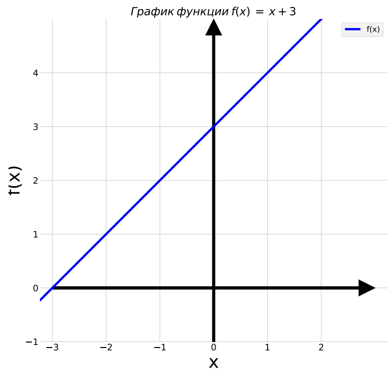
Таблично заданная функция представляет собой список пар значений, где первый элемент пары является аргументом функции, а второй - значение функции.

Функции в математическом анализе играют важную роль в решении различных задач, таких как оптимизация, нахождение экстремумов, решение дифференциальных уравнений и т.д. Они также являются основой для изучения других математических дисциплин, таких как теория вероятностей и статистика.

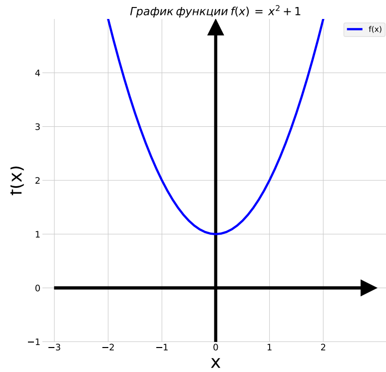
0.5.2 Свойства функций

В математике функции обладают многими свойствами, некоторые из которых важны для их понимания и использования. Ниже перечислены некоторые из основных свойств функций:

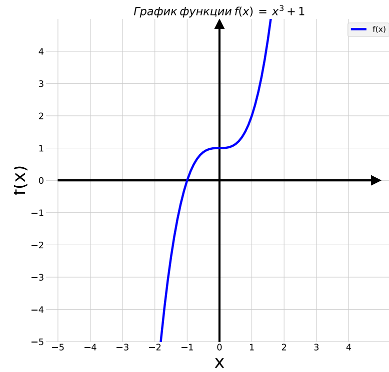
- Однозначность: каждому элементу из области определения функции соответствует единственный элемент из области значений функции.
- Непрерывность: функция называется непрерывной, если изменение значения функции в результате бесконечно малого изменения аргумента также является бесконечно малым.
- Гладкость: функция называется гладкой, если ее производные любого порядка непрерывны.
- Ограниченность: функция называется ограниченной, если ее значения на области определения ограничены константой.
- Монотонность: функция называется монотонной, если она убывает или возрастает на всей области определения или на некотором ее интервале.



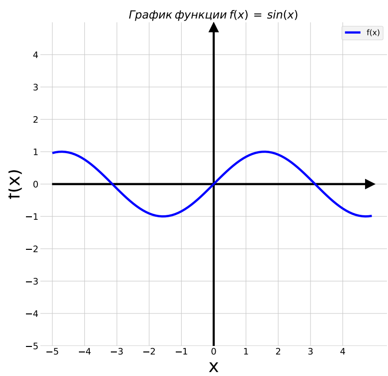
(1) Однозначность функции



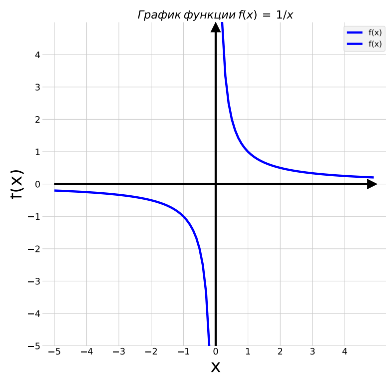
(2) Непрерывность функции



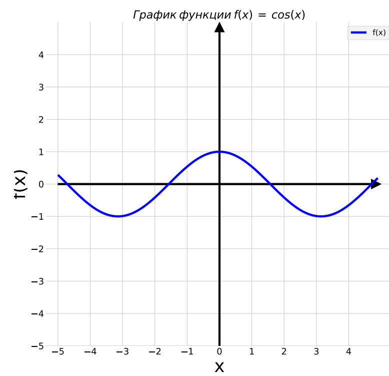
(3) Гладкость функции



(4) Ограниченность функции



(5) Монотонность функции



(6) Периодичность функции

Рисунок 17: Иллюстрация основных свойств функций

- Периодичность: функция называется периодической, если ее значения повторяются через определенный интервал.

Приведем иллюстрацию описанных свойств на примере распространенных функций (Рисунок 17).

0.5.3 Графики часто используемых функций

$f(x) = ax + b$

График функции $f(x) = ax + b$ при a = 1 и и b = 3 (Рисунок 18):
Свойства функции y = ax + b:

- Гладкость: функция y = ax + b гладкая на всей области определения, т.е. не имеет разрывов, углов и изломов.
- Ограниченность: функция y = ax + b не является ограниченной сверху или снизу на всей области определения, за исключением случая, когда a = 0, тогда функция будет ограничена только снизу, если b < 0, или ограничена только сверху, если b > 0.

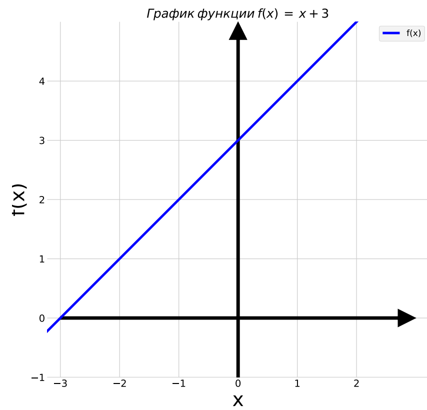


Рисунок 18: $f(x) = ax + b$

- Монотонность: функция y = ax + b может быть монотонной или не монотонной, в зависимости от знака коэффициента a. Если a > 0, то функция возрастает, если a < 0, то функция убывает, а если a = 0, то функция константа и не является монотонной.
- Периодичность: функция y = ax + b не является периодической на всей области определения, за исключением случая, когда a = 0, тогда функция является периодической с любым периодом, проходящим через начало координат.

$f(x) = ax^2 + b$

График функции $f(x) = ax^2 + b$ при a = 1 и и b = 1 (Рисунок 19):

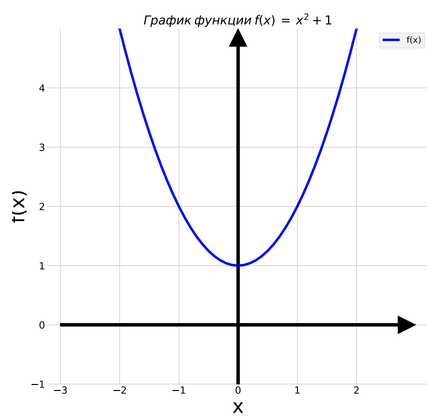


Рисунок 19: $f(x) = ax^2 + b$

Свойства функции $y = ax^2 + b$:

- Гладкость: функция $y = ax^2 + b$ гладкая на всей области определения, т.е. имеет непрерывные производные любого порядка.
- Ограниченность: функция $y = ax^2 + b$ может быть ограниченной сверху или снизу на всей области определения, в зависимости от знака коэффициента a и значения b. Если a > 0, то функция ограничена снизу значением b, а сверху не ограничена. Если a <

0, то функция ограничена сверху значением b, а снизу не ограничена. Если a = 0, то функция является константой и ограничена сверху и снизу значением b.

- Монотонность: функция $y = ax^2 + b$ не является монотонной на всей области определения.
- Периодичность: функция $y = ax^2 + b$ не является периодической на всей области определения.

$f(x) = ax^3 + b$

График функции $f(x) = ax^3 + b$ при a = 1 и и b = 1 (Рисунок 20):

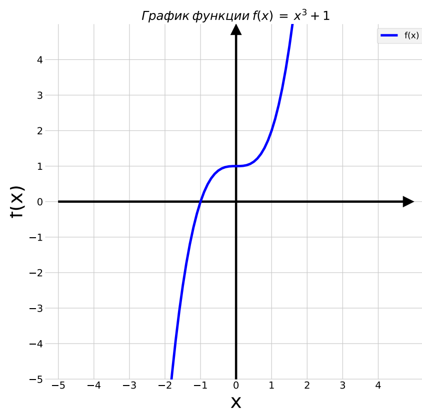


Рисунок 20: $f(x) = ax^3 + b$

Свойства функции $y = ax^3 + b$:

- Гладкость: функция $y = ax^3 + b$ гладкая на всей области определения, т.е. имеет непрерывные производные любого порядка.
- Ограниченность: функция $y = ax^3 + b$ может быть ограниченной сверху или снизу на всей области определения, в зависимости от знака коэффициента a и значения b. Если $a > 0$, то функция ограничена снизу значением b, а сверху не ограничена. Если $a < 0$, то функция ограничена сверху значением b, а снизу не ограничена. Если a = 0, то функция является константой и ограничена сверху и снизу значением b.
- Монотонность: функция $y = ax^3 + b$ может быть монотонной или не монотонной, в зависимости от знака коэффициента a. Если $a > 0$, то функция возрастает на всей области определения, если $a < 0$, то функция убывает на всей области определения, а если a = 0, то функция константа и не является монотонной.
- Периодичность: функция $y = ax^3 + b$ не является периодической на всей области определения.

$f(x) = 1/x$

График функции $f(x) = 1/x$ (Рисунок 21):
Свойства функции $f(x) = 1/x$:

- Гладкость: функция $f(x) = 1/x$ не является гладкой на всей области определения, так как имеет вертикальную асимптоту в точке x=0, где она не определена. Однако, на

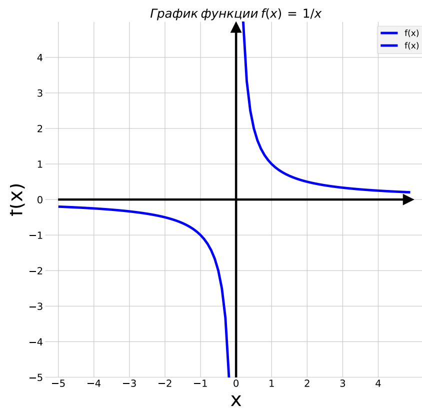


Рисунок 21: $f(x) = 1/x$

любом интервале, не содержащем точку x=0, функция гладкая и имеет непрерывные производные любого порядка.

- Ограниченность: функция $f(x) = 1/x$ не является ограниченной на своей области определения, так как ее значения могут стать очень большими, если x близко к 0. Например, при x=0.0001, значение функции будет равно 10 000, а при x=0.00001 - 100 000. Таким образом, функция не имеет ограничений ни сверху, ни снизу.
- Монотонность: функция $f(x) = 1/x$ является монотонно убывающей на своей области определения, так как при увеличении аргумента x, ее значение убывает. Формально, для любых двух точек x1 и x2 на своей области определения, если x1 < x2, то f(x1) > f(x2).
- Периодичность: f(x) = 1/x не является периодической на своей области определения, так как ее значения зависят только от значения аргумента x и не повторяются через равные интервалы.

$f(x) = \sin(x)$

График функции $(x) = \sin(x)$ (Рисунок 22):

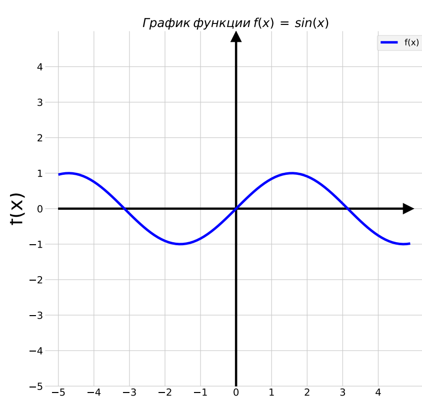


Рисунок 22: $(x) = \sin(x)$

Свойства функции $f(x) = \sin(x)$:

- Гладкость: функция $\sin(x)$ бесконечно дифференцируема на всей числовой оси. Это означает, что ее производные любого порядка существуют везде на числовой оси и непрерывны.
- Ограниченность: функция $\sin(x)$ ограничена сверху и снизу значениями от -1 до 1. Таким образом, ее значения не могут превышать этих границ.
- Монотонность: функция $\sin(x)$ не является монотонной на всей числовой оси, поскольку имеет периодическую структуру. Однако на каждом периоде функция $\sin(x)$ является ограниченной и возрастающей на первой половине периода и убывающей на второй половине периода.
- Периодичность: функция $\sin(x)$ периодическая с периодом 2π. Это означает, что для любого x значение функции $\sin(x)$ будет равно значению функции $\sin(x+2\pi)$, $\sin(x+4\pi)$, $\sin(x-2\pi)$ и т.д.

$f(x) = \cos(x)$

График функции $f(x) = \cos(x)$ (Рисунок 23):

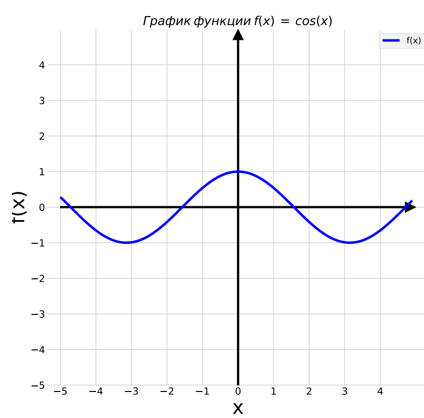


Рисунок 23: $f(x) = \cos(x)$

Свойства функции $f(x) = \cos(x)$:

- Гладкость: функция $\cos(x)$ является бесконечно дифференцируемой на всей числовой оси, что означает, что она гладкая на любом интервале.
- Ограниченность: функция $\cos(x)$ ограничена сверху и снизу значениями от -1 до 1, т.е. $|\cos(x)| <= 1$ для любого x.
- Монотонность: функция $\cos(x)$ не является монотонной на всей числовой оси, так как она пересекает ось x бесконечное число раз. Однако на каждом периоде функция $\cos(x)$ монотонно убывает с увеличением x на первой половине периода, и монотонно возрастает на второй половине.
- Периодичность: функция $\cos(x)$ является периодической с периодом 2π, т.е. $\cos(x+2\pi)$ $= \cos(x)$ для любого x.

$f(x) = e^x$

График функции $f(x) = e^x$ (Рисунок 24):

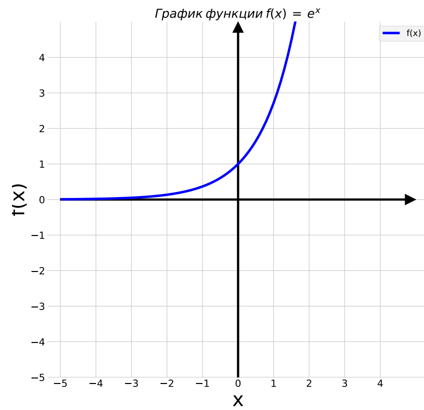


Рисунок 24: $f(x) = e^x$

Свойства функции $f(x) = e^x$:

- Гладкость: функция e^x является бесконечно дифференцируемой на всей числовой оси, что означает, что она гладкая на любом интервале.
- Ограниченность: функция e^x неограничена сверху, но ограничена снизу нулем, т.е. $e^x > 0$ для любого x.
- Монотонность: функция e^x монотонно возрастает на всей числовой оси. Это означает, что с увеличением x значение функции e^x также увеличивается.
- Периодичность: функция e^x не является периодической на всей числовой оси.

$f(x) = ln(x)$

График функции $f(x) = ln(x)$ (Рисунок 25):

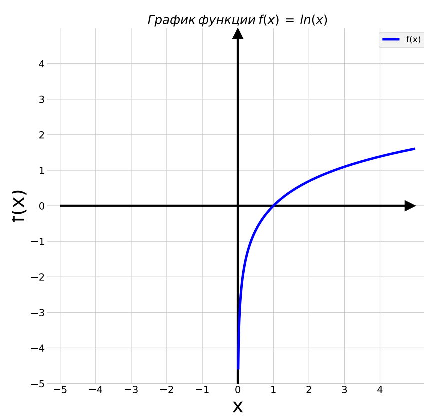


Рисунок 25: $f(x) = ln(x)$

Свойства функции $f(x) = ln(x)$:

- Гладкость: функция $ln(x)$ гладкая на всей области определения (x $>$ 0), то есть она бесконечно дифференцируема.
- Ограниченность: функция $ln(x)$ неограничена сверху, но ограничена снизу нулем, то есть ln(x) $>$ 0 для x $>$ 1 и ln(1) $=$ 0.

- Монотонность: функция $ln(x)$ монотонно возрастает на всей области определения, то есть если x1 < x2, то ln(x1) < ln(x2).
- Периодичность: функция $ln(x)$ не является периодической на всей числовой оси.

0.5.4 Реализация в Python

Реализация основных функций рассмотрена в jupyter notebook файле: lecture_01_code_-labs_03_functions.ipynb

0.5.5 Вопросы для самопроверки

Область значений функции - это ...:

1. множество всех значений аргумента (переменной x)
2. множество всех значений функции (переменной y)
3. объединение множества всех значений аргумента (переменной x) и множества всех значений функции (переменной y)
4. пересечение множества всех значений аргумента (переменной x) и множества всех значений функции (переменной y)

Правильные ответы: 2

Функция называется гладкой, если ...

1. изменение значения функции в результате бесконечно малого изменения аргумента также является бесконечно малым
2. ее значения на области определения ограничены константой
3. ее значения повторяются через определенный интервал
4. ее производные любого порядка непрерывны

Правильные ответы: 4

Какие из приведенных ниже функций являются периодическими?

1. y = a * x$^2 + by = sin(x)$
2. y = cos(x)
3. y = 1/x

Правильные ответы: 2, 3

0.5.6 Резюме по разделу

В этом разделе мы поговорили о функциях. Функция - это правило, которое отображает один набор значений (аргументов) в другой набор значений (значений функции). Функции могут быть определены математически, графически, в виде алгоритмов или программного кода. Некоторые из основных свойств функций:

- Гладкость функции: функция называется гладкой, если ее производные любого порядка непрерывны.

- Ограниченность: свойство функции, которое описывает, насколько функция ограничена в своих значениях.
- Монотонность: свойство функции, которое описывает ее возрастание или убывание при изменении аргумента.
- Периодичность: свойство функции, которое описывает ее повторение с определенным периодом.
- Дифференцируемость: функция является дифференцируемой, если ее производная существует для всех точек в области определения.

Функции широко используются в машинном обучении, в задачах оптимизации, моделирования и прогнозирования. Например, в линейной регрессии функция используется для описания связи между независимыми переменными и зависимой переменной.

0.6 Математический анализ: производные

0.6.1 Определение производной

Производная - это понятие, используемое в математике для описания того, как быстро изменяется функция в каждой точке. Если мы говорим о функции, которая описывает зависимость одной переменной от другой, то производная показывает, как быстро изменяется значение функции с изменением ее аргумента в каждой точке. Один из самых часто встречаемых примеров производной в реальной жизни — это ускорение. Ускорение показывает, как быстро изменяется скорость тела за единицу времени.

Более формально, производная функции определяется как предел отношения приращения функции к приращению ее аргумента при стремлении приращения аргумента к нулю:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

где Δx - бесконечно малое приращение аргумента.

Эта формула показывает, как быстро изменяется функция f(x) в точке x. Если производная положительна, то функция возрастает в этой точке, если отрицательна - функция убывает. Если производная равна нулю, то функция имеет экстремум (максимум или минимум) в этой точке.

Производная имеет много приложений в математике и науке, так как она позволяет описать изменение величин в различных процессах и является ключевым инструментом для решения многих задач.

Существует несколько нотаций (способов записи) производной функции. Наиболее распространенные из них:

- Производная в нотации Лейбница: $\frac{dy}{dx}$. Здесь y - это функция от x, а дробь указывает на приращение функции y по приращению переменной x. Если нужно указать производную более высокого порядка, то можно использовать следующую запись: $\frac{d^n y}{dx^n}$. Здесь n указывает на порядок производной.
- Производная в нотации Лагранжа: $y'(x)$ или $f'(x)$. Эта запись эквивалентна нотации Лейбница $\frac{dy}{dx}$.
- Дифференциал функции: $df = f'(x)dx$. Эта запись означает дифференциал функции $f(x)$, который можно представить как произведение производной $f'(x)$ и бесконечно малого приращения dx.

Выбор записи производной зависит от предпочтений и требований к форматированию в конкретном случае.

Геометрическая интерпретация производной функции заключается в том, что производная в точке определяет скорость изменения функции в этой точке (Рисунок 26). При стремлении приращения аргумента x (красные пунктирные линии) к нулю отношение приращения функции к приращению аргумента будет все точнее отражать скорость роста функции в точке.

В машинном обучении геометрическая интерпретация производной имеет множество применений. Например, она может использоваться в оптимизации функций потерь, которые определяют, насколько хорошо модель машинного обучения соответствует данным. Методы оптимизации, такие как градиентный спуск, используют производную для нахождения минимума функции потерь и оптимизации параметров модели.

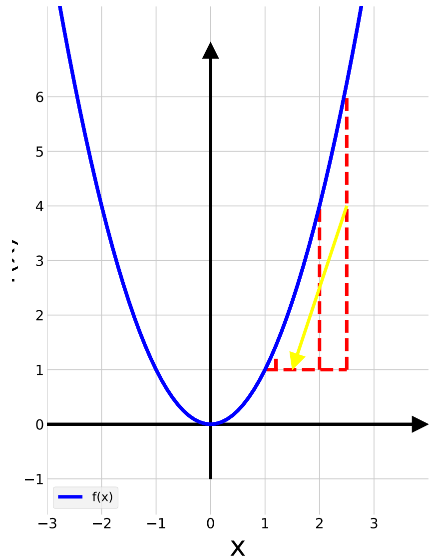


Рисунок 26: Геометрическая интерпретация производной

Если мы рассмотрим график функции $y = f(x)$, то производная функции $f'(x)$ в каждой точке графика показывает наклон касательной к этой точке графика. Если производная положительна, то наклон касательной положительный, что означает, что функция в этой точке возрастает. Если производная отрицательна, то наклон касательной отрицательный, что означает, что функция в этой точке убывает. Если производная равна нулю, то касательная к графику функции является горизонтальной и функция имеет экстремум (максимум или минимум) в этой точке (Рисунок 27):

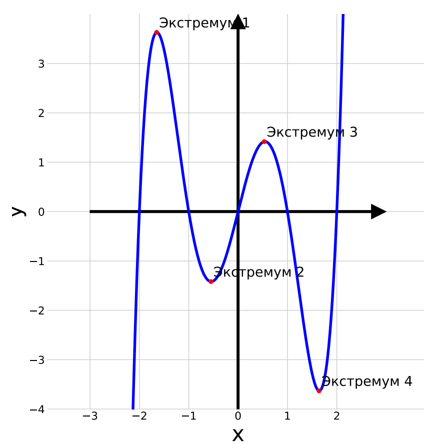


Рисунок 27: Нахождение экстремумов - одна из основных задач производных

Задачи машинного обучения связаны с оптимизацией параметров модели, чтобы минимизировать ошибку на тренировочных данных или максимизировать некоторую функцию потерь. Нахождение экстремумов функций помогает найти оптимальные значения этих параметров.

Если график функции $y = f(x)$ имеет касательную в точке $(a,f(a))$, то производная функции в этой точке равна тангенсу угла наклона касательной к графику в этой точке. Пусть m - угол наклона касательной к графику функции $y = f(x)$ в точке $(a,f(a))$. Тогда, если мы рассмотрим малый прирост аргумента dx и соответствующий прирост функции df, то получим следующее:

$$\tan m = \lim_{dx \to 0} \frac{df}{dx}$$

Если существует производная функции $f(x)$ в точке a, то этот предел существует и равен значению производной $f'(a)$, т.е.

$$\tan m = \frac{df}{dx}$$

Таким образом, производная функции в точке касания графика касательной к этому графику равна тангенсу угла наклона касательной в этой точке.

0.6.2 Недифференцируемые функции

Недифференцируемая функция - это функция, которая не имеет производной в какой-либо точке своей области определения. То есть, производная функции не существует в этой точке, либо существует, но не является конечной.

Примеры недифференцируемых функций

Любая функция, в которой есть углы, будет недифференцируема в точке нахождения угла. Функция модуля знака $|x|$ является недифференцируемой в нуле, так как ее производная не существует в этой точке. Если попытаться провести касательную, она будет определена неоднозначно (Рисунок 28):

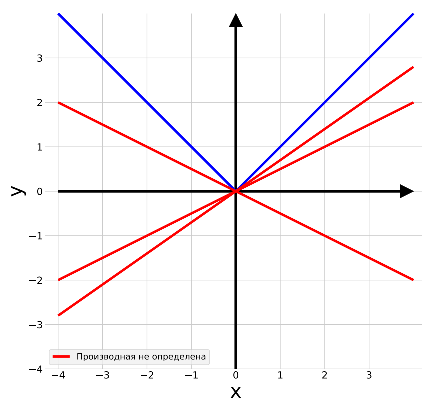


Рисунок 28: Функция $|x|$ недифференцируема в точке 0

Разрывные функции — функции, имеющая разрыв в некоторых точках — также недифференцируемы в этих точках (Рисунок 29):

Недифференцируемые функции могут вызывать некоторые проблемы при решении математических задач и приложений. Некоторые из проблем, связанных с недифференцируемыми функциями, включают:

- Ограничение области определения: некоторые задачи требуют дифференцируемых функций, и использование недифференцируемых функций может ограничить область определения их применения.

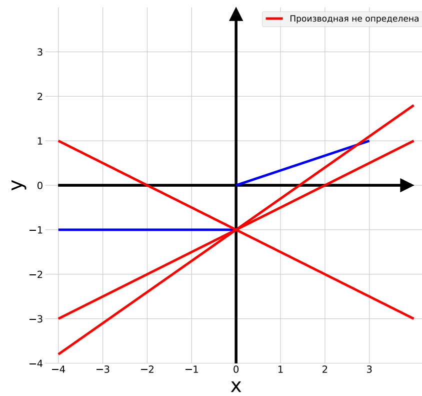


Рисунок 29: Разрывная (кусочно-линейная) функция недифференцируема в точке 0

- Сложность вычислений: недифференцируемые функции могут быть более сложными в вычислении, так как их производные могут иметь особенности, такие как разрывы или бесконечные значения.
- Несуществование градиента: в некоторых задачах оптимизации требуется нахождение градиента функции, и недифференцируемые функции могут быть проблемой, так как градиент не существует в некоторых точках.
- Необходимость использования других методов: некоторые задачи могут требовать анализа недифференцируемых функций, и для их решения может потребоваться использование других методов, например, методов интегрального и функционального анализа.

Функции являются недифференцируемыми в точках, где тангенс имеет вертикальные асимптоты. Это происходит потому, что в таких точках производная функции не существует. Например, функция тангенса имеет вертикальные асимптоты в точках $\frac{\pi}{2} + k\pi$, где k - любое целое число. В этих точках тангенс стремится к бесконечности, и производная функции не существует(Рисунок 30):

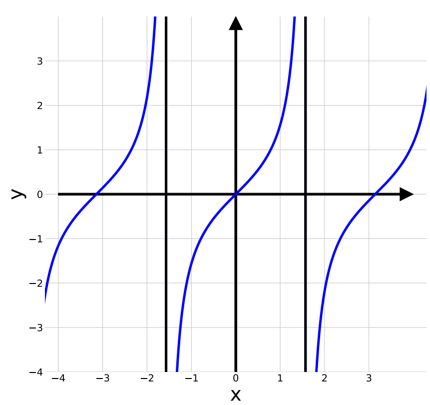


Рисунок 30: Разрывная (кусочно-линейная) функция недифференцируема в точке 0

Соответственно, функция является недифференцируемой, если:

- имеются углы

- имеются разрывы
- имеются точки с вертикальными асимптотами тангенса

0.6.3 Правила дифференцирования

Основные правила дифференцирования:

- Сумма производных:
$$(f(x) + g(x))' = f'(x) + g'(x)$$
- Разность производных:
$$(f(x) - g(x))' = f'(x) - g'(x)$$
- Произведение производных:
$$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x)$$
- Частное призводных:
$$\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - f(x)g'(x)}{[g(x)]^2}$$
- Производная произведения функции на константу равна произведению константы и производной функции:
$$(cf(x))' = cf'(x)$$
- Производная константы равна нулю:
$$\frac{d}{dx}[f(x) + C] = f'(x)$$
- Правило цепочки: производная композиции функций[1] равна произведению производной внешней функции и производной внутренней функции:
$$(f(g(x)))' = f'(g(x))g'(x)$$
  Это правило играет важную роль в машинном обучении. В нотации Лейбница оно может быть записано следующим образом:
$$\frac{df}{dx} = \frac{df}{dg} \cdot \frac{dg}{dx}$$
  Эта формула показывает, что производная сложной функции $f(g(x))$ равна произведению производной внешней функции $f'(g(x))$ и производной внутренней функции $g'(x)$. Например, если $f(x) = \sin(x^2)$ и $g(x) = x^2$, то
$$\frac{df}{dx} = \frac{d}{dx}\sin(g(x)) = \cos(g(x)) \cdot \frac{dg}{dx} = \cos(x^2) \cdot 2x$$
  Это правило может быть использовано для дифференцирования более сложных функций, которые представляют собой композицию нескольких функций.

Эти правила могут быть расширены и использованы в более сложных случаях, но они образуют основу дифференцирования функций.

[1]Композиция функций, или сложная функция — операция, при которой результат одной функции используется в качестве аргумента для другой функции

0.6.4 Производные часто используемых функций

Ниже приведены производные часто используемых функций:

- Константа:
$$f'(C) = 0$$
- Линейная функция:
$$f'(ax + b) = a$$
- Степенная функция:
$$f'(x^n) = nx^{n-1}$$
- Степенная функция с отрицательным целым показателем:
$$f'(x^{-n}) = -nx^{-n-1}$$
- Тригонометрические функции:
$$f'(\sin(x)) = \cos(x)$$
$$f'(\cos(x)) = -\sin(x)$$
$$f'(\tan(x)) = \sec^2(x)$$
Обратные тригонометрические функции:
$$f'(\arcsin(x)) = \frac{1}{\sqrt{1 - x^2}}$$
$$f'(\arccos(x)) = -\frac{1}{\sqrt{1 - x^2}}$$
$$f'(\arctan(x)) = \frac{1}{1 + x^2}$$
- Экспоненциальная функция:
$$f'(e^x) = e^x$$
- Логарифмическая функция:
$$f'(\ln(x)) = \frac{1}{x}$$

Эти производные могут быть использованы для нахождения производных более сложных функций, которые представляют собой композицию нескольких функций.

0.6.5 Реализация в Python

Реализация основных производных рассмотрена в jupyter notebook файле: lecture_01_-code_labs_04_derivatives.ipynb

0.6.6 Вопросы для самопроверки

Какие из перечисленных ниже определений производной являются верными?

1. наклон касательной к графику функции в данной точке.
2. показатель скорости изменения функции в данной точке.
3. отношение значения аргумента к значению функции.
4. отношение изменения функции к изменению ее аргумента при бесконечно малом изменении аргумента.

Правильные ответы: 1, 2, 4

Функция является недифференцируемой, если:

1. имеются точки с вертикальными асимптотами тангенса
2. имеются разрывы
3. имеются углы
4. значения функции на порядки превосходят значения аргумента

Правильные ответы: 1, 2, 3

Какие из приведенных ниже нотаций являются устоявшимися формами записи производной функции:

1. нотация Лейбница
2. нотация Лагранжа
3. нотация Гаусса
4. нотация Маркова

Правильные ответы: 1, 2

0.6.7 Резюме по разделу

В этом разделе мы поговорили о производных. Производная функции - это показатель скорости изменения значения функции в каждой ее точке. Математически производная функции f(x) в точке x_0 определяется как предел отношения изменения значения функции f(x) к изменению ее аргумента x при стремлении изменения аргумента к нулю:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

где Δx - бесконечно малое приращение аргумента (приращение, стремящееся к нулю)

Производные функций широко используются в машинном обучении. Например, производные могут использоваться для оптимизации функций потерь при обучении нейронных сетей. Оптимизация функций потерь требует нахождения экстремумов функции, что может быть сделано путем вычисления производных и использования методов оптимизации, таких как градиентный спуск. Производные также могут использоваться для нахождения экстремумов функций при обучении моделей машинного обучения, таких как линейная регрессия и логистическая регрессия.

0.7 Резюме по модулю

В этом модуле мы рассмотрели такие темы, как основные термины в области искусственного интеллекта и машинного обучения, библиотеки Python, используемые в рамках курса, математический аппарат, используемый в машинном обучении, векторы в и матрицы в линейной алгебре, функции и производные в математическом анализе.

Мы рассмотрели основные термины, которые широко используются в машинном обучении, включая искусственный интеллект, машинное обучение, глубинное обучение, наука о данных, анализ данных, обучение с учителем, обучение без учителя и обучение с подкреплением. Кроме того, мы ознакомились с основными специальными терминами, такими как объекты, признаки, целевая переменная, набор данных, модель, функция потерь, классификация, регрессия, кластеризация, снижение размерности.

Мы обсудили концепцию векторов. В математике вектор - это объект, который имеет определенную длину и направление в пространстве, и может быть представлен в виде упорядоченного набора чисел, называемых компонентами вектора. Свойства векторов включают длину, направление, операции суммы и разности, скалярное и векторное произведения, а также угол между векторами. В машинном обучении векторы используются для представления признаков или наблюдений в виде чисел. Например, вектор может представлять цвет изображения в RGB-формате, частоты звука или другие характеристики данных. Алгоритмы машинного обучения используют векторы для выполнения операций, таких как классификация, кластеризация и регрессия. Кроме того, векторы могут использоваться для решения задач оптимизации.

Мы изучили понятие матриц в контексте машинного обучения. Матрица - это таблица, состоящая из строк и столбцов, представляющая собой прямоугольный массив чисел. Некоторые основные свойства матриц:

- Размерность: количество строк и столбцов определяет размерность матрицы.
- Операции: матрицы можно складывать, вычитать, умножать на число и другие матрицы.
- Транспонирование: можно транспонировать матрицу, заменив строки на столбцы и наоборот.
- Обратная матрица: некоторые матрицы имеют обратную, которая используется для решения систем линейных уравнений.
- Определитель: число, вычисляемое для квадратной матрицы, может использоваться для решения систем линейных уравнений и для определения обратной матрицы.
- Ранг матрицы: максимальное число линейно независимых строк или столбцов в матрице.

Мы обсудили понятие функций. Функция - это правило, которое преобразует один набор значений (аргументы) в другой набор значений (значения функции). Функции могут быть определены математически, графически, в виде алгоритмов или программного кода. Некоторые из основных свойств функций:

- Гладкость функции: функция называется гладкой, если ее производные любого порядка непрерывны.
- Ограниченность: свойство функции, которое описывает, насколько функция ограничена в своих значениях.
- Монотонность: свойство функции, которое описывает ее возрастание или убывание при изменении аргумента.

- Периодичность: свойство функции, которое описывает ее повторение с определенным периодом.
- Дифференцируемость: функция является дифференцируемой, если ее производная существует для всех точек в области определения.

Мы рассмотрели концепцию производных. Производная функции представляет собой индикатор скорости изменения значения функции в каждой ее точке при стремлении изменения аргумента к нулю:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

где Δx - бесконечно малое приращение аргумента.

Производные функций находят широкое применение в области машинного обучения. Например, они используются для оптимизации функций потерь при обучении нейронных сетей.