18+

Samael Blackart

АНАТОМИЯ НЕЙРОСЕТЕЙ

Тайные коды искусственного разума

Samael Blackart

Анатомия нейросетей. Тайные коды искусственного разума

«Издательские решения»

Blackart S.

Анатомия нейросетей. Тайные коды искусственного разума / S. Blackart — «Издательские решения»,

ИИ — не магия, а статистика. Но чтобы выжимать из нейросетей максимум, нужно знать, как устроен их «двигатель». Эта книга — проводник под капот искусственного интеллекта: от биологического нейрона до трансформера. Вы поймете, почему ИИ галлюцинирует, как работают CNN, RNN и Self-Attention, и напишете первую сеть на Python. Только архитектура, код и архетипы машинного разума. Станьте архитектором ИИ, а не просто пользователем.

Содержание

Анатомия нейросетей
Тайные коды искусственного разума

Samael Blackart

ISBN 978-5-0070-7357-8
Создано в интеллектуальной издательской системе Ridero

АНАТОМИЯ НЕЙРОСЕТЕЙ
Тайные коды искусственного разума

Архитектура Внутреннего Мира: Квантовая Психология. Книга 2
Автор: Samael Blackart

ДИСКЛЕЙМЕР

Данное издание носит исключительно образовательный, исследовательский и культурологический характер. Материалы представлены как объект изучения истории культуры, психологии архетипов, символизма и антропологии сознания.

Автор не занимается пропагандой каких-либо религиозных, мистических или эзотерических учений. Все описанные практики, символы и концепции рассматриваются исключительно в контексте:

— Историко-культурного анализа
— Психологии архетипов
— Когнитивной науки и нейробиологии восприятия
— Символической антропологии

Автор не гарантирует каких-либо конкретных результатов от применения описанных методик. Материалы не являются заменой профессиональной психологической, медицинской или юридической помощи. Все упоминания исторических личностей, мифологических персонажей и культурных артефактов приводятся исключительно в научно-исследовательских целях.

Все упоминания торговых марок (PyTorch, TensorFlow, GPT, Claude, Gemini, BERT и др.) приведены в научно-исследовательских и образовательных целях и принадлежат их правообладателям.

Материалы не являются заменой профессионального технического образования.

Введение
Заглянем под капот божества

В первой книге серии мы разобрались с иллюзиями. Вы поняли, что ИИ — не волшебник, а статистическая машина. Вы узнали, почему он врёт, и написали свой первый код для бенчмарка разных моделей.

Но это всё — взгляд снаружи. Вы видели, как ИИ работает. Но вы не видели, как он устроен.

Это всё равно что водить машину, не зная, как работает двигатель. Вы можете ехать из точки А в точку Б. Но когда машина сломается (а она сломается — в виде галлюцинаций, странных ответов, неоптимальных результатов), вы не сможете её починить. Вы даже не поймёте, что именно сломалось.

Эта книга — про двигатель.

Мы заглянем под капот искусственного интеллекта. Не чтобы стать инженерами-нейротехниками (хотя если захотите — сможете). А чтобы понять архитектуру настолько, чтобы осознанно выбирать инструменты, отлаживать проблемы и выжимать из ИИ максимум.

Главный тезис этой книги: тот, кто понимает архитектуру нейросетей, программирует ИИ в 10 раз эффективнее того, кто этого не понимает.

Это не преувеличение. Это факт, подтверждённый практикой. Когда вы знаете, что трансформер работает через механизм внимания, вы по-другому формулируете промпты. Когда вы понимаете, как свёрточные сети видят изображения, вы иначе ставите задачи компьютерного зрения. Когда вы знаете, почему рекуррентные сети забывают длинный контекст, вы не удивляетесь, что ИИ «забыл» начало вашего длинного документа.

Знание архитектуры — это не академическая роскошь. Это практический инструмент.

Как построена эта книга

Книга разделена на 4 части, каждая со своей целью:

Часть I. Фундамент (главы 1—3)

Здесь мы разберёмся с основами: что такое нейрон, перцептрон, многослойная сеть. Это база, без которой дальше идти бессмысленно. Написано так, что поймёт даже школьник.

Часть II. Архитектуры (главы 4—7)

Ядро книги. Четыре главные архитектуры нейросетей: полносвязные, свёрточные, рекуррентные, трансформеры. Для каждой — история, принцип работы, сильные и слабые стороны, код.

Часть III. Обучение (главы 8—10)

Как ИИ учится? Что такое градиентный спуск? Почему данные важнее алгоритмов? Что такое переобучение и как с ним бороться? Без этого раздела вы не поймёте, почему одни модели работают, а другие нет.

Часть IV. Практика (главы 11—12)

Пишем свою первую нейросеть на Python. Разбираемся, как выбрать архитектуру под конкретную задачу.

Архетипический подход

В этой книге мы будем использовать особый приём: каждую архитектуру нейросетей мы рассмотрим через архетип. Не как украшение, а как инструмент понимания.

Архетипы — это универсальные паттерны, которые наш мозг использует для обработки информации. Когда мы говорим «трансформер — это архетип Внимания», это не метафора. Это способ зацепить новую информацию за уже существующие в вашем мозге структуры.

Четыре архетипа четырёх архитектур:

— Перцептрон — Архетип Воина (простой, прямой, бьёт в одну цель)

— Свёрточная сеть — Архетип Зрения (видит паттерны, ищет структуры)

— Рекуррентная сеть — Архетип Памяти (хранит прошлое, связывает события)

— Трансформер — Архетип Внимания (фокусируется на важном, игнорирует шум)

Этот подход — часть фирменного стиля «Гримуара Апейрона». Он помогает не просто запомнить факты, а понять суть.

Для кого эта книга

Если вы:
— Прочитали Книгу 1 и хотите углубиться
— Программист, который хочет понять, как работают нейросети
— Предприниматель, который хочет осознанно выбирать ИИ-решения
— Студент, изучающий машинное обучение
— Просто любознательный человек, которому интересно, как устроен ИИ
— эта книга для вас.

Мы не будем грузить вас математикой. Формулы будут, но только там, где они действительно нужны, и только в приложениях. Основной текст — на языке аналогий, примеров и кода.

Но если вы математик и хотите увидеть формулы — в Приложении А есть математический минимум для каждой темы.

Что вы получите после прочтения

После прочтения этой книги вы:
— Будете понимать, как устроены нейросети изнутри
— Сможете объяснить разницу между CNN, RNN и трансформером
— Напишете свою первую нейросеть на Python
— Сможете осознанно выбирать архитектуру под задачу
— Будете понимать, почему ИИ ошибается, и как это исправить
— Перестанете быть пользователем ИИ — станете его архитектором
Готовы заглянуть под капот? Тогда начинаем.

Часть I. Фундамент

Глава 1
От биологии к математике: как родился нейрон

Прежде чем говорить об искусственных нейронах, нужно понять, откуда они взялись. А взялись они из биологии — из настоящего нейрона, клетки вашего мозга.

Эта глава — про мост между биологией и математикой. Про то, как учёные попытались описать работу мозга языком чисел. И как из этой попытки родился искусственный интеллект.

Биологический нейрон: прототип всего

Ваш мозг состоит из примерно 86 миллиардов нейронов. Каждый нейрон — это клетка, которая умеет принимать сигналы, обрабатывать их и передавать дальше.

Структура биологического нейрона:

— Дендриты — «входы» нейрона. Принимают сигналы от других нейронов.

— Тело нейрона (сома) — обрабатывает сигналы.

— Аксон — «выход» нейрона. Передаёт сигнал дальше.

— Синапсы — соединения между нейронами. Сила соединения определяет, насколько сильно один нейрон влияет на другой.

Как это работает:

1. Нейрон получает сигналы через дендриты от других нейронов.
2. Каждый сигнал имеет свою «силу» (частоту импульсов).
3. Нейрон суммирует все входящие сигналы.
4. Если суммарный сигнал превышает определённый порог — нейрон «срабатывает» и посылает импульс по аксону.
5. Этот импульс через синапсы передаётся следующим нейронам.

Всё. Это и есть вся «логика» биологического нейрона: суммирование + порог.

Просто? Да. Но из 86 миллиардов таких простых элементов рождается сознание, мышление, память, эмоции. Это один из величайших парадоксов природы.

Первая математическая модель: 1943 год

В 1943 году нейробиолог Уоррен Макаллок и математик Уолтер Питтс предложили первую математическую модель нейрона. Они спросили: можно ли описать работу биологического нейрона языком математики?

Ответ: да.

Они предложили простую модель:

— Нейрон получает N входов: x1, x2, …, xN

— Каждый вход имеет свой «вес» (силу влияния): w1, w2, …, wN

— Нейрон вычисляет взвешенную сумму: S = w1*x1 + w2*x2 + … + wN*xN

— Если S превышает порог T — нейрон выдаёт 1 (срабатывает)

— Если S меньше T — нейрон выдаёт 0 (не срабатывает)

Это и есть первый искусственный нейрон. Модель Маккалока-Питтса.

Пример из жизни:

Представьте, что вы решаете, идти ли сегодня на прогулку. У вас есть несколько факторов:

— Погода хорошая (x1 = 1, если да; 0, если нет)

— Есть свободное время (x2 = 1 или 0)

— Нет головной боли (x3 = 1 или 0)

Но факторы не равнозначны. Погода для вас важнее всего (w1 = 0.5). Время — менее важно (w2 = 0.3). Головная боль — средне (w3 = 0.4).

Считаем:

S = 0.5*1 +0.3*1 +0.4*1 = 1.2

Если ваш порог T = 1.0, то 1.2> 1.0, и вы идёте на прогулку.

Если погода плохая (x1 = 0):

S = 0.5*0 +0.3*1 +0.4*1 = 0.7

0.7 <1.0 — вы остаётесь дома.

Вот и вся модель. Входы + веса + суммирование + порог.

Это поразительно просто. И это поразительно мощно. Потому что из таких простых элементов можно построить систему, которая умеет решать сложные задачи.

От биологии к математике: что потеряли, что приобрели

Что мы потеряли при переходе от биологии к математике:

— Биологическую сложность (химические процессы, тысячи типов нейронов, нелинейную динамику)

— Энергоэффективность (мозг потребляет 20 Вт, современные нейросети — мегаватты)

— Способность учиться на нескольких примерах (мозгу достаточно одного-двух примеров)

Что мы приобрели:

— Математическую строгость (можем доказывать теоремы о поведении сети)

— Масштабируемость (можем строить сети с миллиардами параметров)

— Воспроизводимость (одна и та же сеть даёт одинаковый результат)

— Скорость (электронные нейроны работают быстрее биологических)

Это классический инженерный компромисс: мы жертвуем универсальностью ради специализации. Биологический нейрон — универсал. Искусственный нейрон — специалист. Но миллионы специалистов, работающих вместе, могут больше, чем один универсал.

Почему это важно понимать

Зачем вам знать про биологический нейрон?

Во-первых, чтобы понимать ограничения ИИ. Искусственные нейросети — это очень упрощённая модель биологических. Они не умеют того, что умеет мозг: учиться на одном примере, переносить знания между задачами, осознавать себя. Когда вы это понимаете, вы не ждёте от ИИ невозможного.

Во-вторых, чтобы понимать возможности ИИ. Несмотря на упрощение, искусственные нейросети решают задачи, которые раньше считались исключительно человеческими: распознавание образов, перевод языков, игра в шахматы, написание текстов. Понимание, как это работает, даёт вам власть над инструментом.

В-третьих, чтобы говорить на одном языке с разработчиками. Когда вы слышите «веса», «порог», «активация» — вы понимаете, о чём речь. Это делает вас не просто пользователем, а партнёром в разработке ИИ-решений.

Практическое задание

Задание 1.1: Ваш личный нейрон
Придумайте собственную задачу принятия решения. Например:
— Смотреть ли новый фильм
— Заказывать ли пиццу
— Покупать ли новую книгу
Определите:
— Какие входы (факторы) вы учитываете? (минимум 3)
— Какие веса у этих факторов? (сумма весов не важна)
— Какой у вас порог принятия решения?
Проверьте модель на 5 реальных ситуациях. Работает ли она? Где даёт сбой?

Глава 2
Перцептрон: атом искусственного разума

В предыдущей главе мы построили математическую модель нейрона. Теперь дадим ей имя: перцептрон.

Перцептрон — это простейшая искусственная нейронная сеть. Один нейрон. Одна задача. Но именно на перцептроне построено всё современное машинное обучение. Как атом — на материи.

В этой главе мы:

— Разберём перцептрон до винтика

— Напишем его на Python с нуля

— Поймём, что он умеет, а что нет

— Увидим его архетип — Воин

Архетип Воина: простота и сила

Перцептрон — это архетип Воина.

Воин прост. У него одна цель. Он бьёт прямо, без обходных путей. Он не думает о нюансах — он принимает решение: да или нет, атака или отступление.

Перцептрон точно такой. Он решает только один тип задач: бинарную классификацию. Да или нет. 1 или 0. Кошка или не кошка. Спам или не спам.

Но не путайте простоту с примитивностью. Воин — это не глупец. Это тот, кто довёл простое до совершенства. И перцептрон — это не примитив. Это чистая математическая идея, доведённая до кристальной ясности.

Устройство перцептрона

Перцептрон состоит из:

— Входов: x1, x2, …, xN

— Весов: w1, w2, …, wN

— Смещения (bias): b

— Сумматора: S = w1*x1 + w2*x2 + … + wN*xN + b

— Функции активации: y = f (S)

Входы (x1, x2, …, xN):

Это данные, которые поступают в перцептрон. Для задачи распознавания спама входами могут быть: «есть ли слово „выигрыш“ в письме?», «есть ли ссылки?», «отправитель в чёрном списке?» и т. д.

Веса (w1, w2, …, wN):

Каждый вход имеет свой вес. Вес показывает, насколько важен этот вход для принятия решения. Если вес большой — вход сильно влияет на решение. Если вес близок к нулю — вход почти не важен. Если вес отрицательный — вход «толкает» решение в противоположную сторону.

Смещение (bias, b):

Это «базовая склонность» перцептрона. Даже если все входы нулевые, смещение может «подтолкнуть» перцептрон к решению.

Функция активации:

Превращает сумму в выходное значение. Для перцептрона это обычно ступенчатая функция:

— Если S> = 0, то y = 1
— Если S <0, то y = 0

Геометрическая интерпретация

Перцептрон можно представить геометрически.

Представьте двумерное пространство (два входа: x1 и x2). Каждая точка в этом пространстве — это набор входов.

Перцептрон проводит через это пространство прямую линию: w1*x1 + w2*x2 + b = 0.

Все точки по одну сторону линии — класс 1. Все точки по другую сторону — класс 0.

Задача обучения перцептрона — найти такую линию, которая правильно разделит точки двух классов.

Это и есть линейный классификатор. Перцептрон умеет решать только линейно разделимые задачи.

Критический момент: перцептрон НЕ может решить задачу XOR (исключающее ИЛИ). Это доказал Минский в 1969 году, и это доказательство «убило» интерес к нейросетям на 15 лет.

Почему? Потому что для XOR нельзя провести одну прямую, которая отделит класс 1 от класса 0. Нужна как минимум двухслойная сеть. Об этом — в следующей главе.

Код: перцептрон с нуля на Python

Давайте напишем перцептрон с нуля. Без библиотек машинного обучения. Чистая математика.

```
# Перцептрон с нуля
# Архетип: Воин
import numpy as np

class Perceptron:
«"»
Простейший искусственный нейрон.
Решает задачи бинарной классификации.
«"»
def __init__ (self, input_size, learning_rate=0.01):
«"»
Инициализация перцептрона.
Параметры:
— input_size: количество входов
— learning_rate: скорость обучения
«"»
# Веса — случайные числа от -1 до 1
self. weights = np. random. randn (input_size)
# Смещение — ноль
self.bias = 0.0
# Скорость обучения
self.lr = learning_rate

def activate (self, x):
```

```
«"»
Ступенчатая функция активации.
Возвращает 1, если x> = 0, иначе 0.
«"»
return 1 if x> = 0 else 0

def predict (self, inputs):
```

Конец ознакомительного фрагмента.