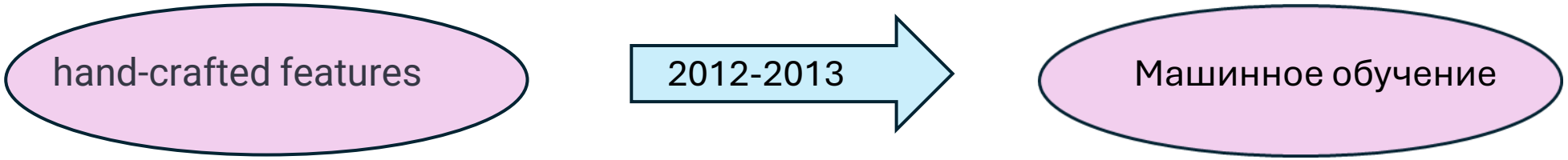
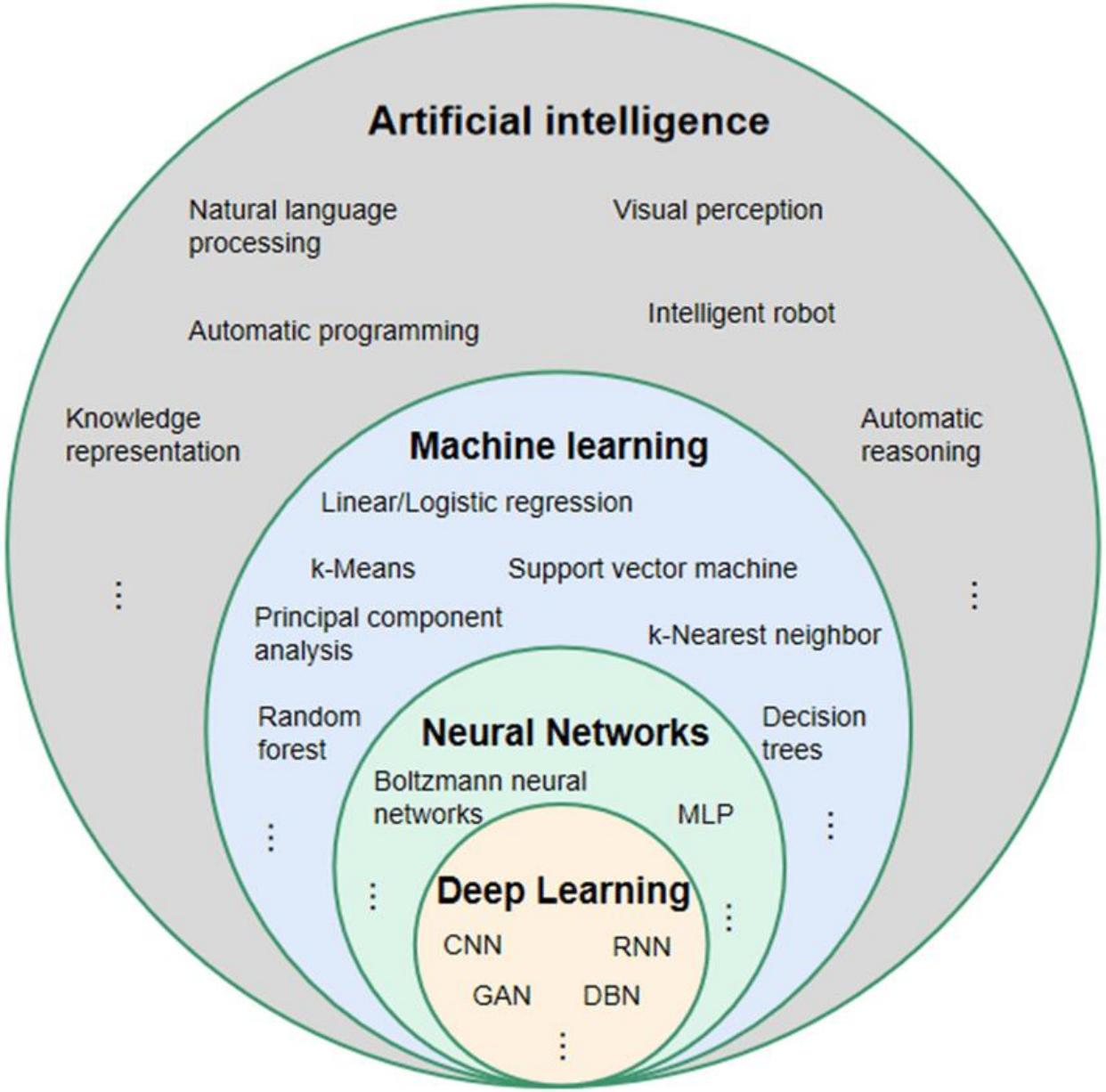
Архитектуры нейронных сетей в компьютерном зрении



плохо масштабируемы и очень чувствительны к изменениям

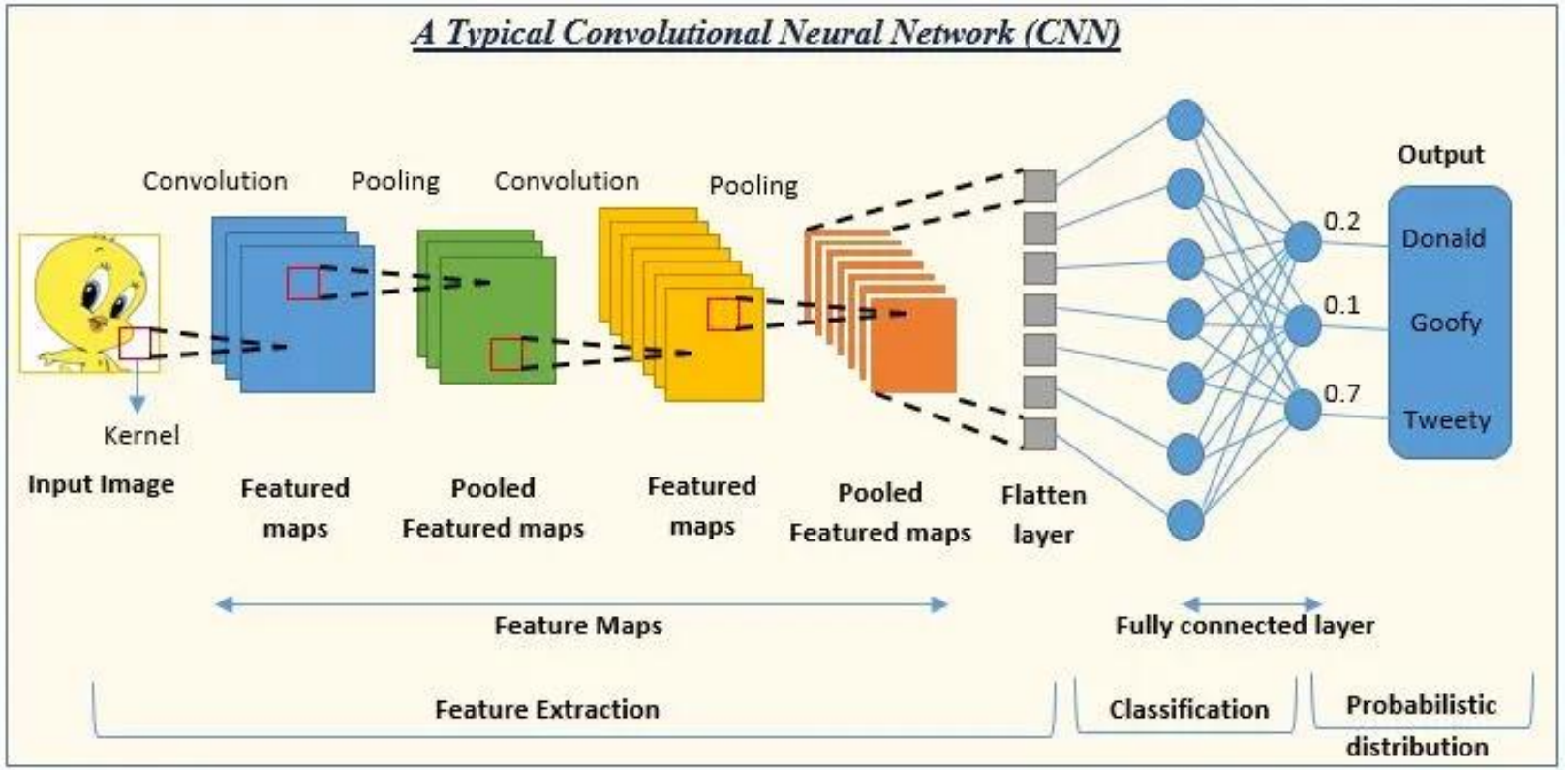
требовательны к ресурсам

Переломный момент произошел в 2012 году на конференции NeurIPS,
когда была представлена модель под названием AlexNet,
которая существенно обошла всех конкурентов
в рамках соревнования по классификации изображений на 1000 классов —
ImageNet Large Scale Visual Recognition Challenge (ILSVRC)



Место нейронных сетей во всем множестве алгоритмов, основанных на данных

Ключевым преимуществом нейронных сетей перед других алгоритмами как ML, так и не ML стало автоматическое обучение признаков, на основе которых принимается решение



Визуализация классической архитектуры НС, в которой выделен блок нахождения признаков и принятие решения на их основе

Основные сферы деятельности — машинное обучение, компьютерное зрение, мобильная робототехника и вычислительная нейробиология.

Лауреат премии Тьюринга (2018, совместно с Бенжио и Хинтоном за формирование направления глубокого обучения).

Получил докторскую степень по информатике в Университете Пьера и Марии Кюри в 1987 году.

Я. Лекун

8 июля 1960 (64 года)

Известен работами по применению нейросетей к задачам оптического распознавания символов и машинного зрения.

Разработал серию методов машинного обучения, в том числе свёрточные нейронные сети.

Один из основных создателей технологии сжатия изображений DjVu.

Вместе с Леоном Боту создал язык программирования Lush.

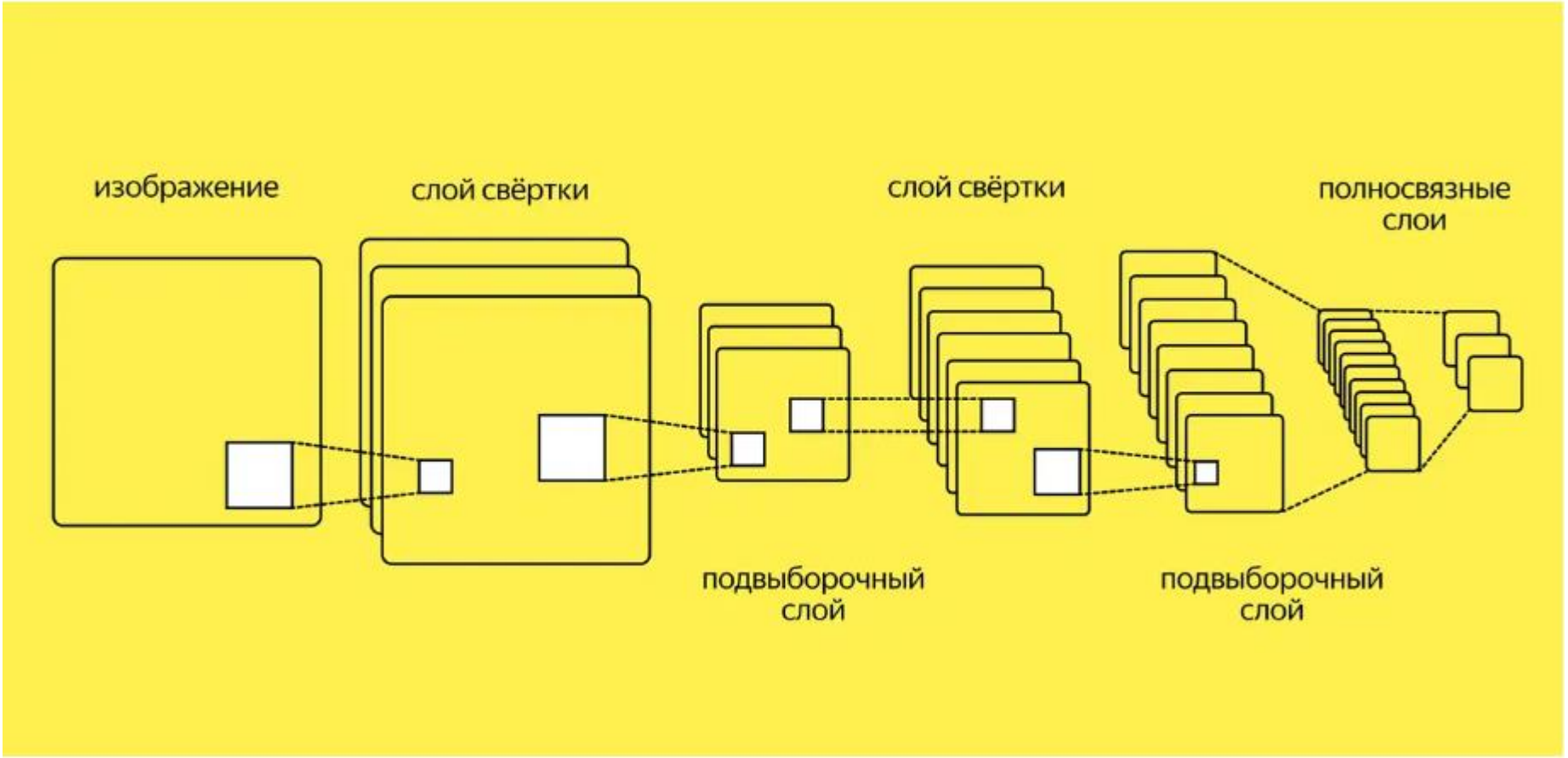
Для реализации механизма автоматическое обучение признаков Я. Лекуном были придуманы свёрточные слои, которые основаны на идее функции свёртки, которая, в свою очередь, очень часто используется в классических алгоритмах компьютерного зрения (БПФ, Гауссово размытие и т.п.).

В последующем они были частично заменены на блоки внимания (self-attention).

Базовые понятия из областей

- линейной алгебры (перемножения матриц, скалярное произведение и т.п.)
- теории вероятностей (случайная переменная, функция вероятности и т.п.)
- математического анализа (предел, производная, дифференцирование сложной функции, градиент)
- машинного обучения (обучение с учителем, без учителя, функция потерь и т.п.)
- компьютерного зрения (свертка, преобразование Фурье, размытие и т.п.)

Структура свёрточной нейросети

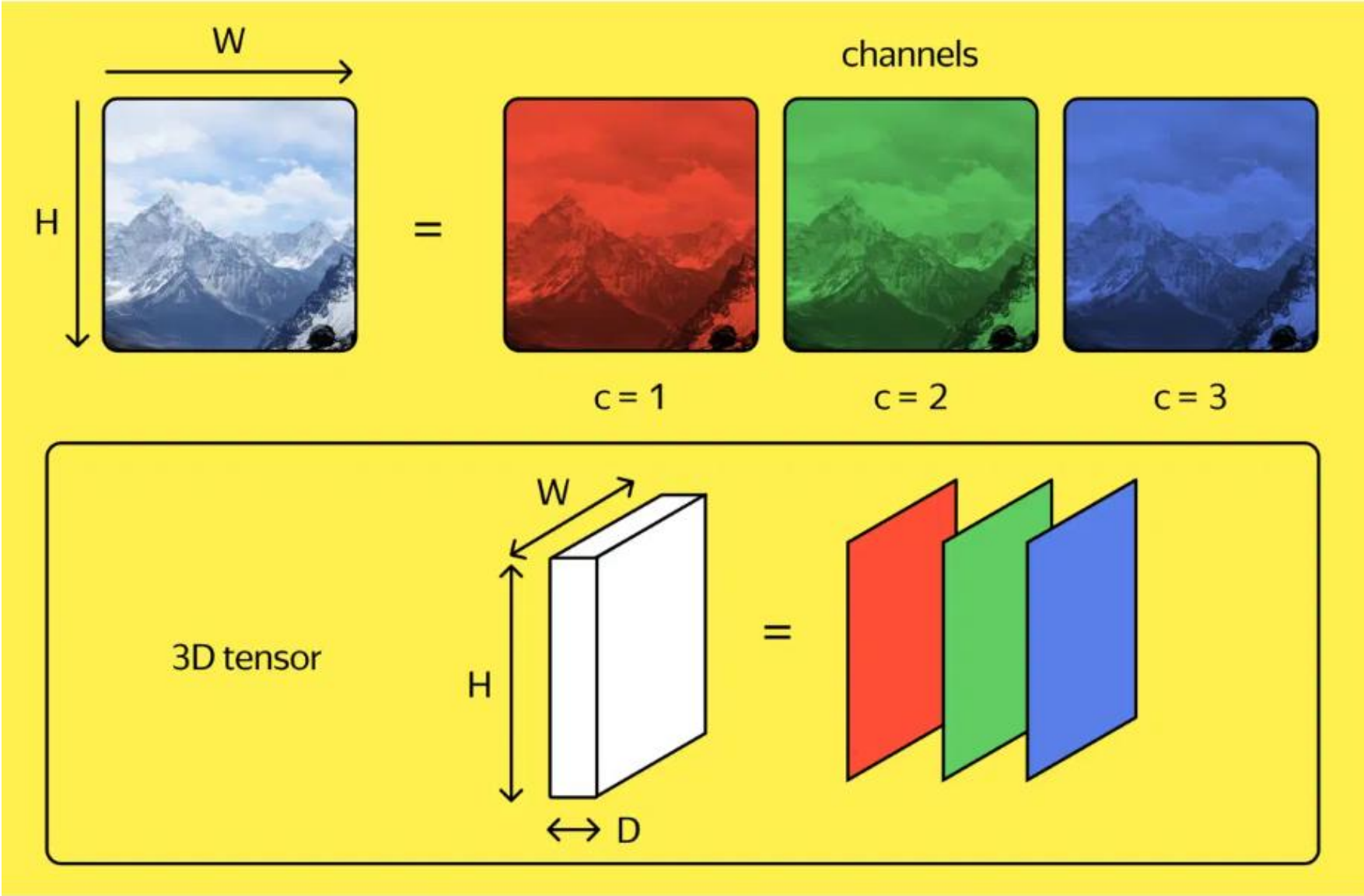


Свёртка и пулинг чередуются несколько раз, чтобы выделять всё более сложные признаки

Входные данные

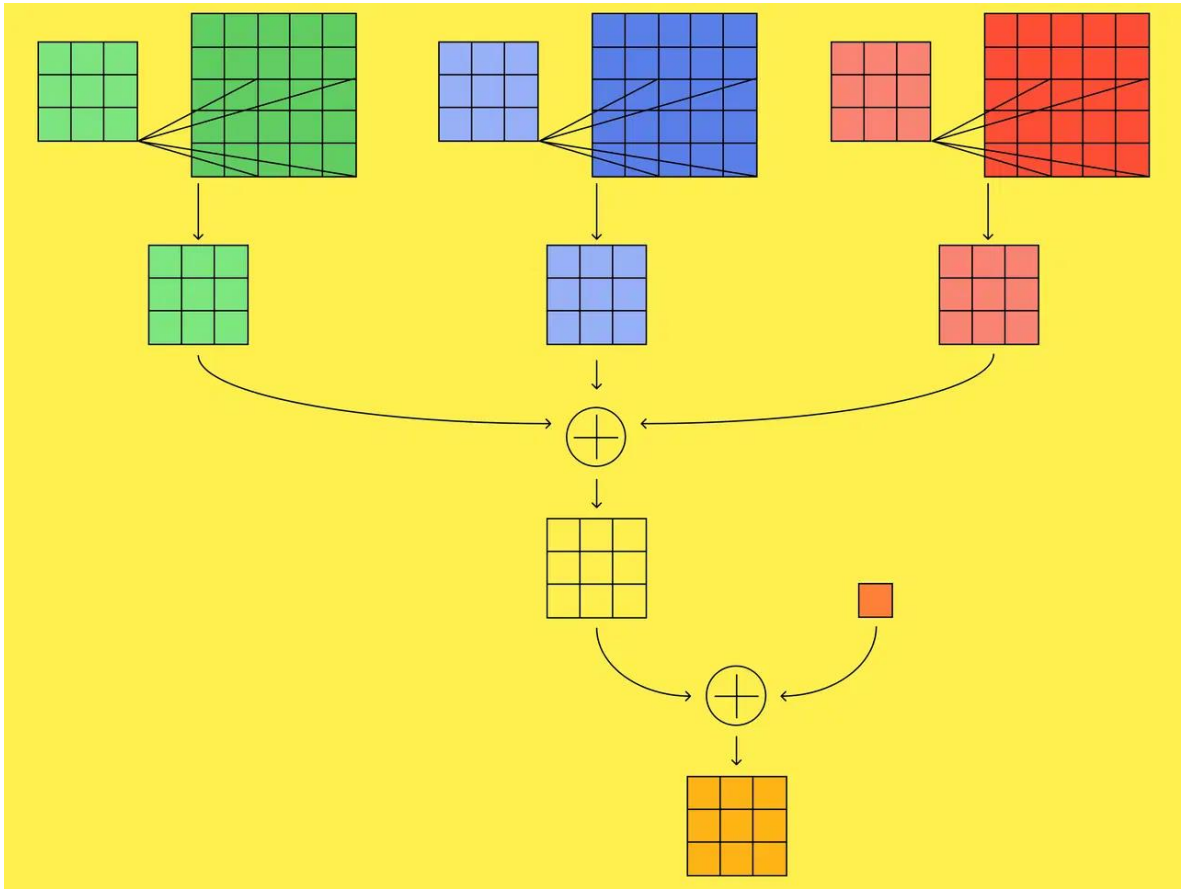
Нейросети в основном обучаются на цветных изображениях в формате RGB.

Свёрточная нейронная сеть «видит» изображение в особом представлении — в виде трёхмерных массивов чисел или массивов матриц. В математике это называется тензорами.



Операция свёртки по всему изображению

- Фильтр проходит по каждому пикселю изображения. На выходе получается новая матрица.
- Числа полученных матриц суммируются в одну матрицу.
- К каждому значению матрицы добавляется одинаковое число — значение, на которое переместился фильтр, или шаг свёртки.
  Шаг равен 1 — фильтр перемещался на один пиксель.
  Шаг равен 2 — фильтр шагал на 2 пикселя.
  Финальная матрица — это один канал выходной карты признаков.
- Все каналы, или матрицы, которые получили после обработки изображения фильтрами, объединяются в один тензор. В итоге получается изображение другого размера и с другим числом каналов.



Количество фильтров в слое зависит от числа признаков.

Основные элементы свёрточной нейронной сети

- свёрточный слой,
- пулинг,
- нормализация по батчу — пакетная нормализация (англ. batch-normalization),
- полносвязный слой.

Свёрточные нейронные сети состоят из нескольких слоёв.
Чем больше слоёв, тем мощнее архитектура и лучше обучение нейросети.

Свёрточный или конволюционный слой

Свёрточный или конволюционный слой является небольшим фильтром, который скользит по изображению, преобразуя его в новое изображение (того же или меньшего размера)

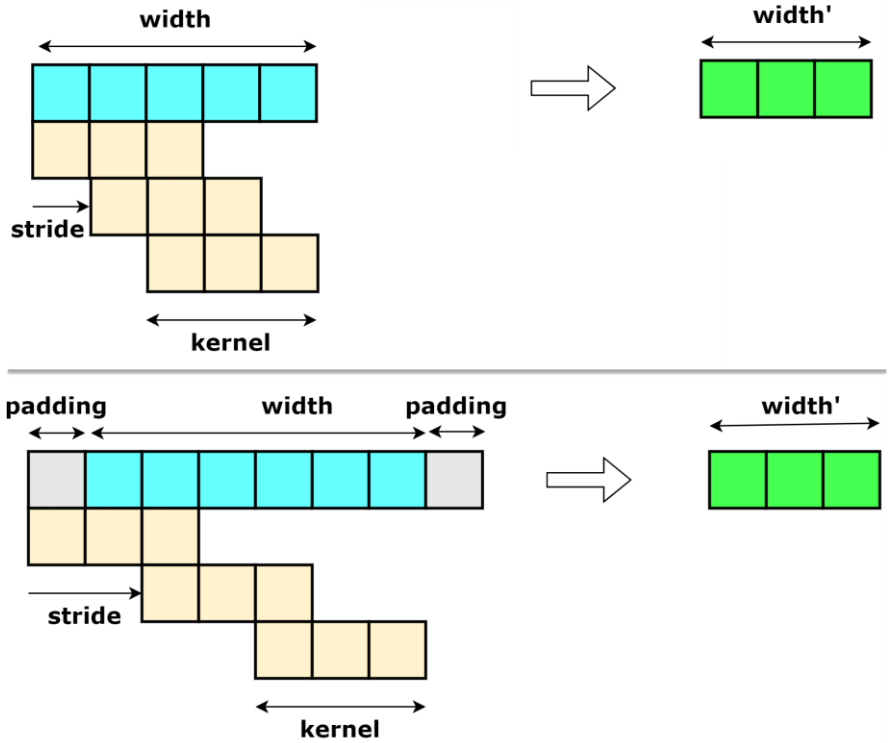
Формула для ширины (и аналогичная для высоты) результирующего изображения:

width' = int((width + 2*padding - kernel)/stride + 1),

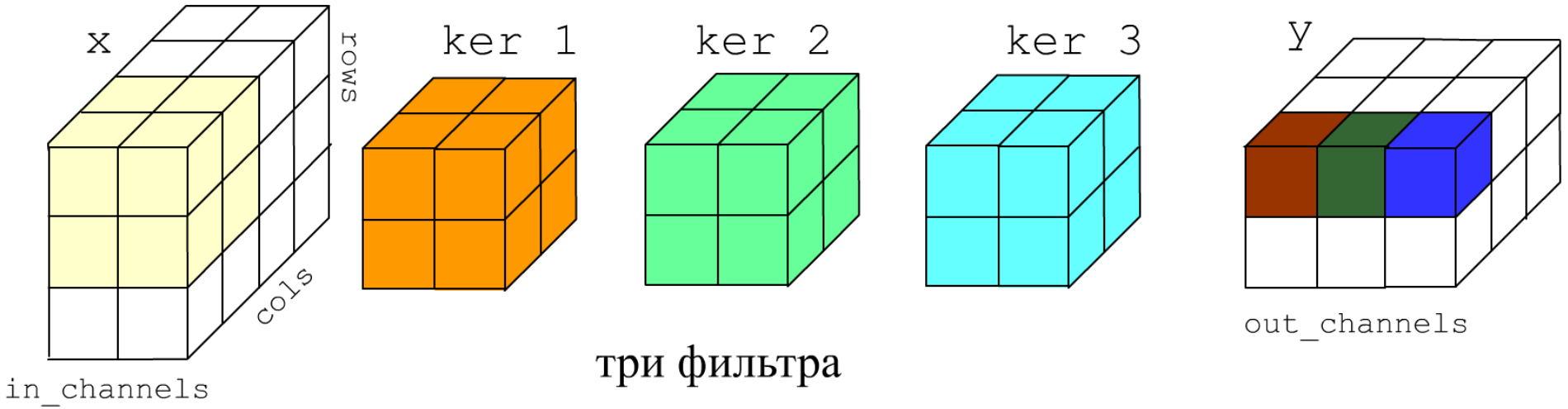
- padding - ширина в пикселях рамки слева и справа от изображения,
- kernel - ширина ядра
- stride шаг с которым он скользит по изображению

(на верхнем рисунке stride=1, padding=0,
а нижнем stride=2, padding=1 и в обоих случаях kernel=3).

Если stride=1, то чтобы размеры изображения не изменилось для kernel = 3, 5, 7, ..., нужен padding = 1, 2, 3,...



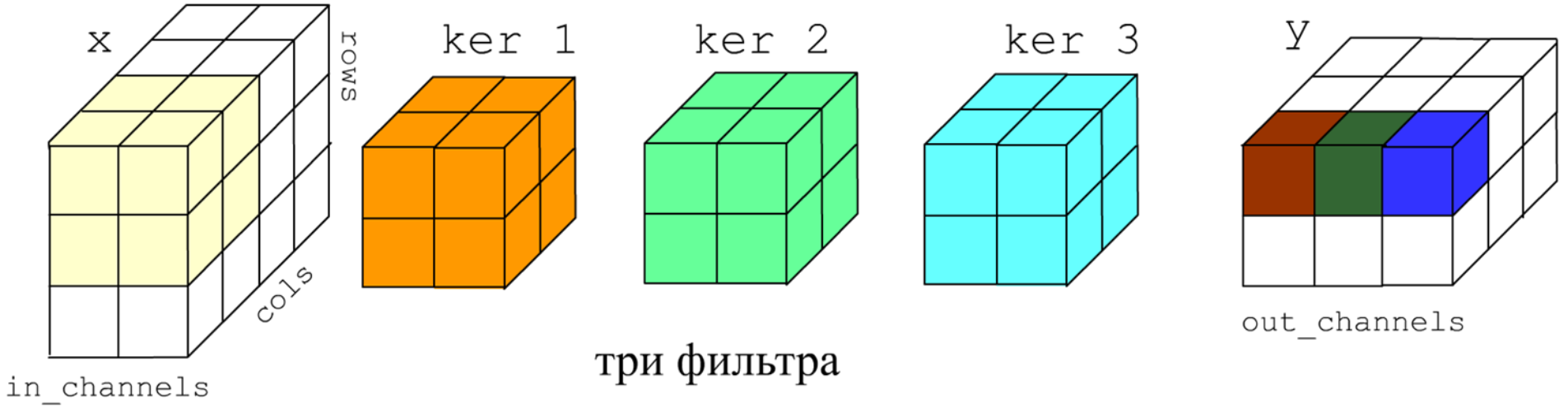
Фильтрация многоканальных изображений



Конволюционый слой на выходе может иметь произвольное число каналов.
Пример: на вход слоя поступает два канала, а на выходе получается три.

Для каждого выходного канала формируется обучаемая 3D матрица параметров (плюс смещение).
Каждая из них независимо и на всю глубину производит вычисление результаты работы такого 3D фильтра.

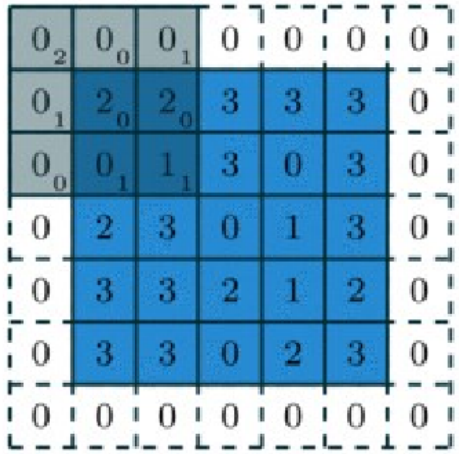
Ключевые параметры

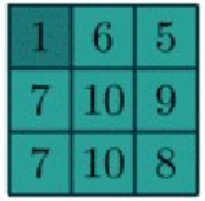


Таким образом, при создании свёрточного слоя **ключевыми параметрами** являются:

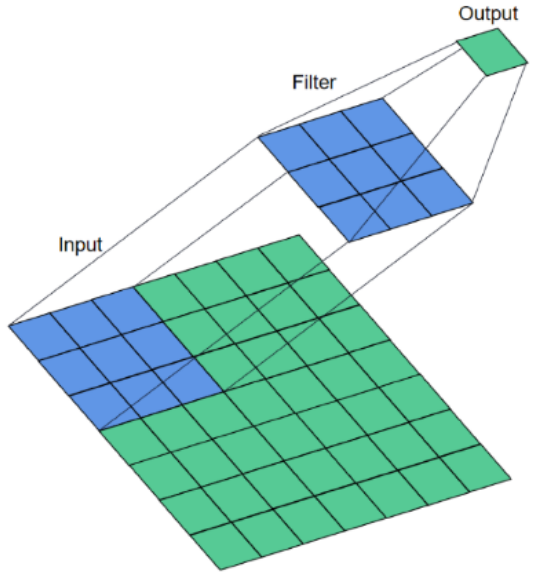
- число входных каналов (глубина фильтров),
- число выходных каналов (количество фильтров),
- размер ядра (ширина и высота фильтров),
- шаг stride с которым фильтр скользит по стопке изображений (входных каналов)

Параметры заполнения и расширения

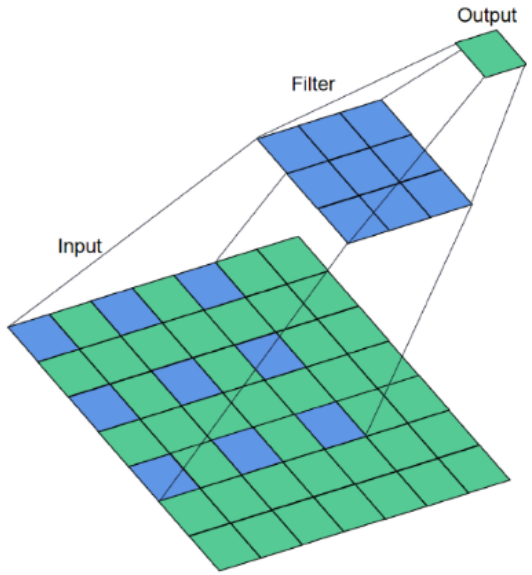




`padding = 1`
`padding_mode = 'zeros'`



`dilation = 1`



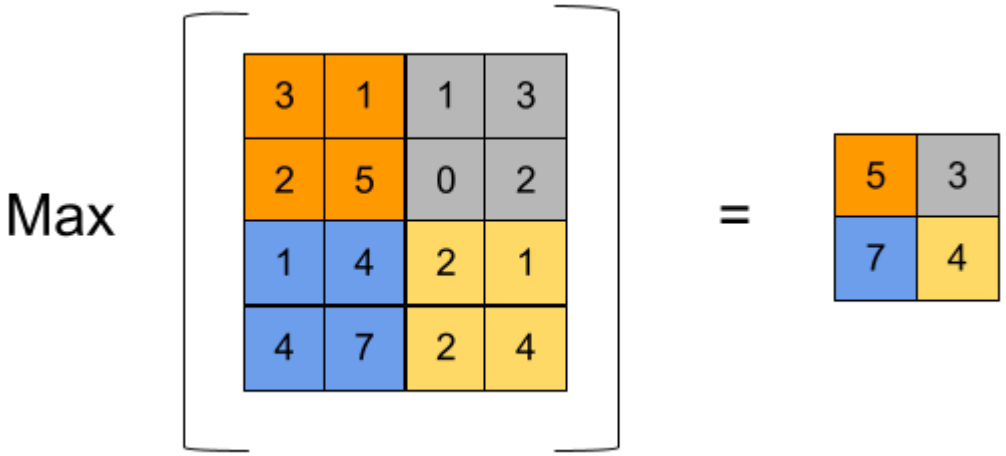
`dilation = 2`

Если мы хотим, чтобы при конволюции размер изображения не менялся, следует окружить его рамкой из "фейковых" пикселей. Для ядра 3 следует взять padding = 1, для ядра 5 - padding = 2 и т.д.

Пулинг max pooling

Второй ключевой составляющей свёрточных сетей является слой пулинг, в частности max pooling.
Он вычисляет максимальное значение пикселя на входном канале внутри своего ядра

Как и конволюционный слой он имеет размер ядра и шаг.
Число каналов на выходе всегда совпадает с числом каналов на входе.
Максимальное значение вычисляется внутри каждого входного канала независимо.
Свёрточный слой перемешивает все входные каналы, а пулинг этого не делает.
Слой пулинга не содержит обучаемых параметров.



Кроме уменьшения размера карты признаков (ширины и высоты стопки каналов),
слой пулинга выделяет важные признаки (с максимальным значением).
К тому же он делает сеть более устойчивой к небольшим сдвигам изображения (в пределах ядра).

Пулинг AvgPool

Реже используется AvgPool, который работает аналогично MaxPool, но при этом вычисляет среднее значение пикселей данного канала, которые попадают в ядро.

AdaptiveAvgPool полностью эквиваленте AvgPool, но, вместо указания размера ядра, принимает форму желаемого выхода. По полученному входу, он автоматически подбирает необходимое ядро

Уменьшение размеров

Отметим, что сужение не обязательно делать при помощи слоя MaxPool2d.

Если stride фильтра в Conv, например, равно 2, то выходные изображения будут в 2 раза меньше,

а если не использовать заполнение (padding), то на каждой конволюции будет "откусываться" периметр карты.

Пакетная нормализация

Пакетная нормализация (англ. batch-normalization) — метод, который позволяет повысить производительность и стабилизировать работу искусственных нейронных сетей.

Нормализация входного слоя нейронной сети обычно выполняется путем масштабирования данных, подаваемых в функции активации.

Например, когда есть признаки со значениями от 0 до 1 и некоторые признаки со значениями от 1 до 1000, то их необходимо нормализовать, чтобы ускорить обучение.

Нормализацию данных можно выполнить и в скрытых слоях нейронных сетей, что и делает метод пакетной нормализации.

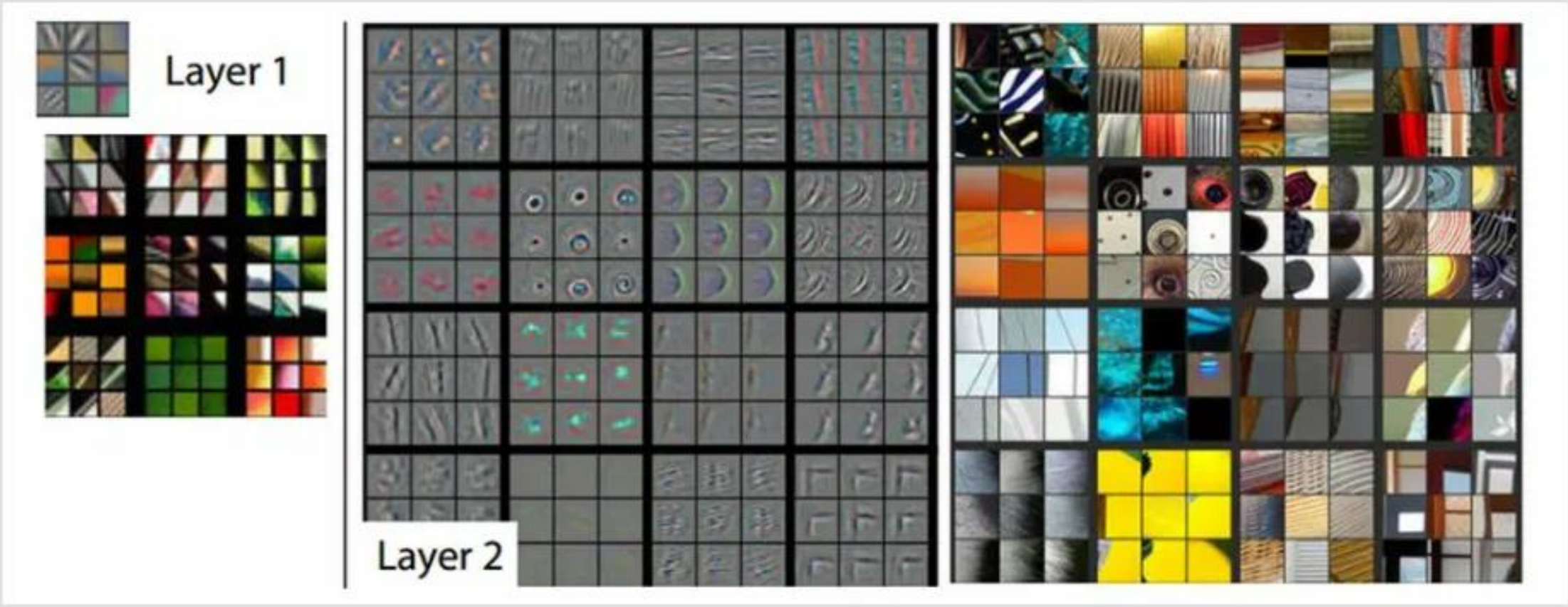
Где встраивать пакетную нормализацию?

Где встраивать пакетную нормализацию — вопрос интуиции и экспериментов.

Например:

- В полносвязных сетях с несимметричной активационной функцией (ReLU, Sigmoid) её стоит ставить после неё.
- С симметричными функциями (Tanh) - перед.

Первые слои нейросети

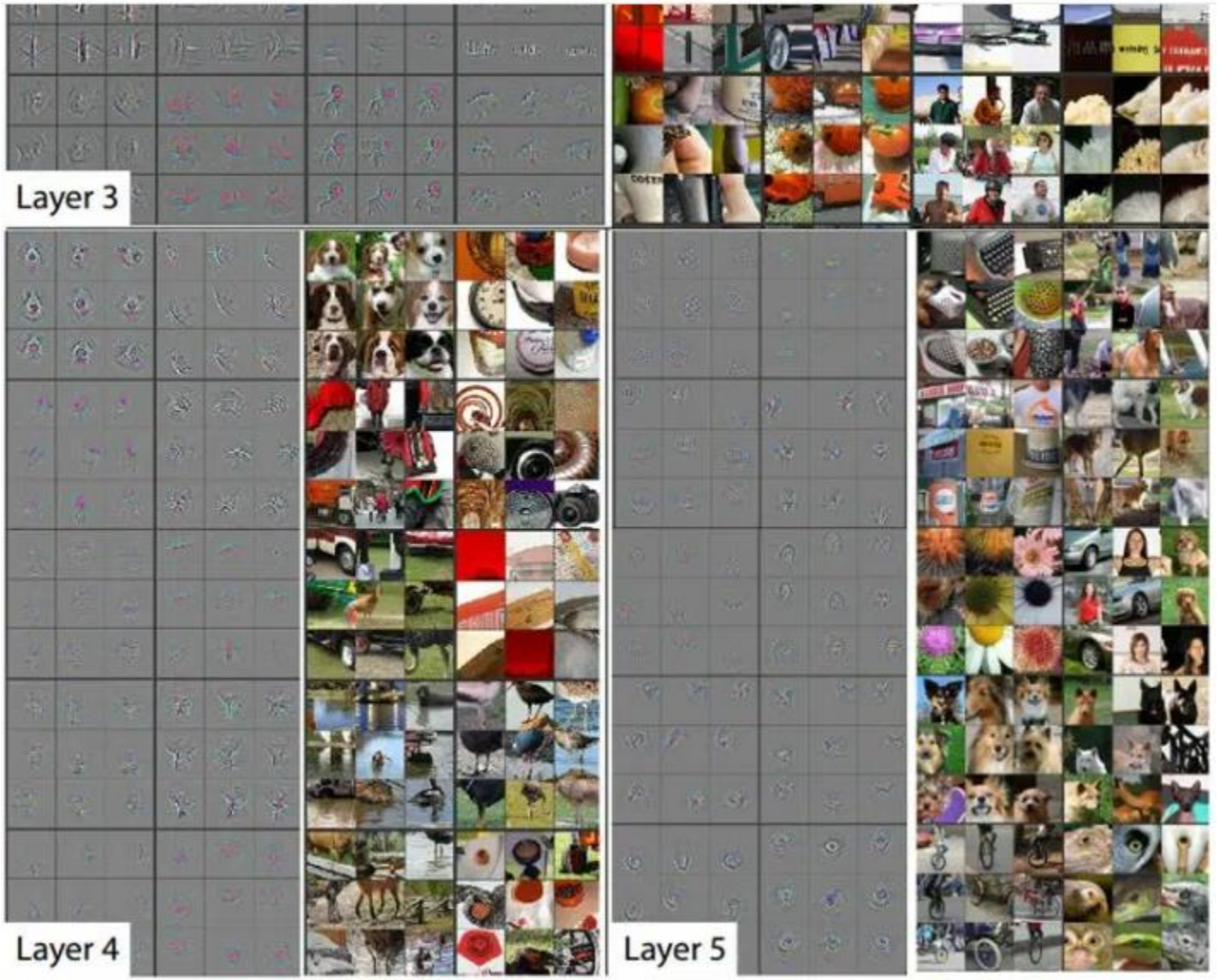


Здесь она ищет границы объектов по перепадам света. Пример от Adit Deshpande

Сложные признаки,

которые получились на последних слоях, нейросеть будет классифицировать с помощью полносвязного слоя, чтобы найти ответ на вопрос, что изображено на картинке, сова или кошка.

На последних слоях нейросеть пытается понять, что или кто перед ней.
Например, есть что-то похожее на колёса. Возможно, на фотографии велосипед. Но это не точно, поэтому альтернативный вариант — глаза птиц.
Пример от Adit Deshpande



Архитектуры свёрточных нейронных сетей

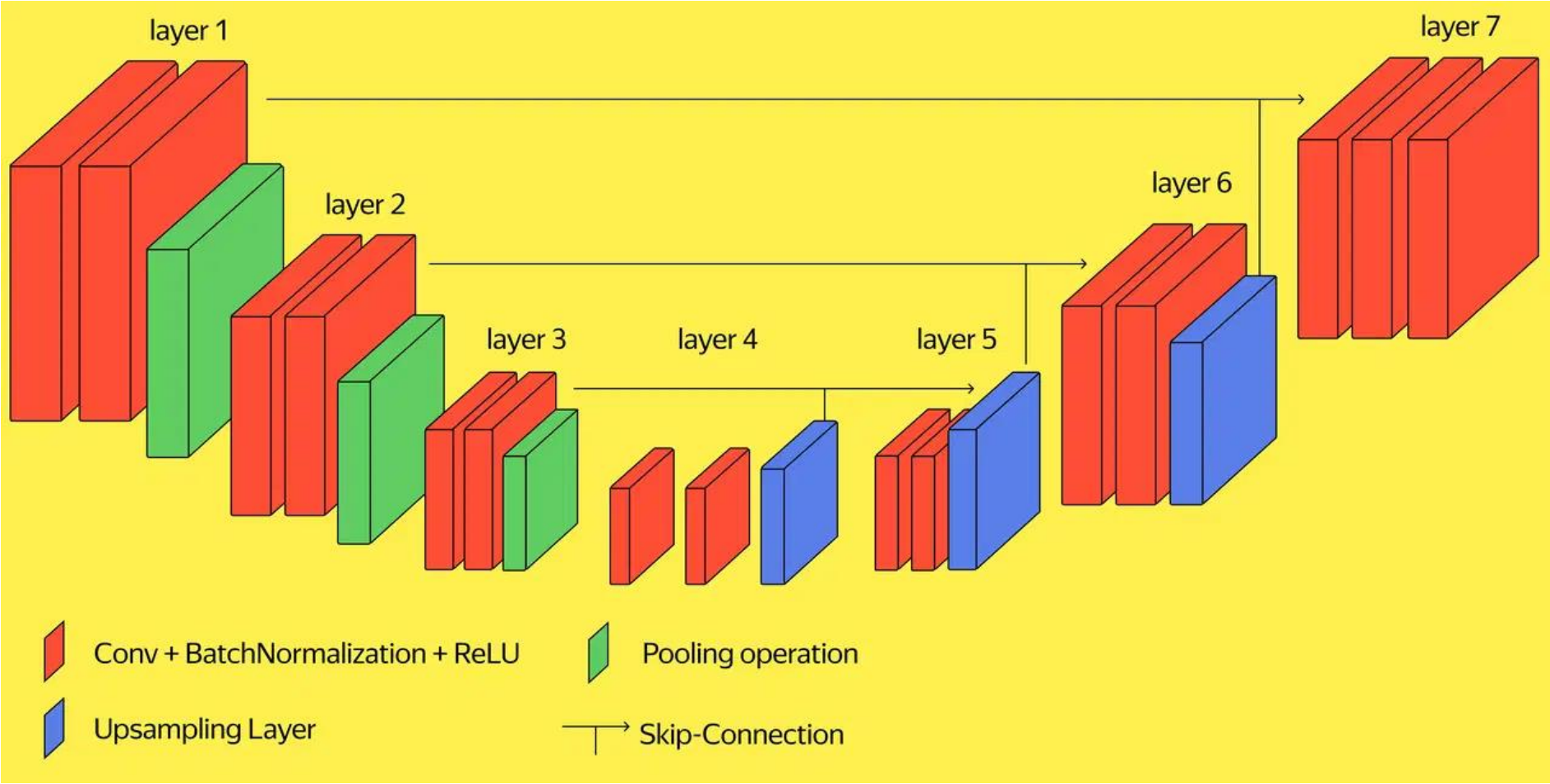
Нагляднее всего архитектуру свёрточных нейросетей видно на двух примерах — ResNet и U-Net

Они похожи, но используются для разных задач

- ResNet — для классификации,
- а U-Net — для сегментации

Архитектура свёрточной сети U-Net

состоит из нескольких слоёв, некоторые из которых чередуются



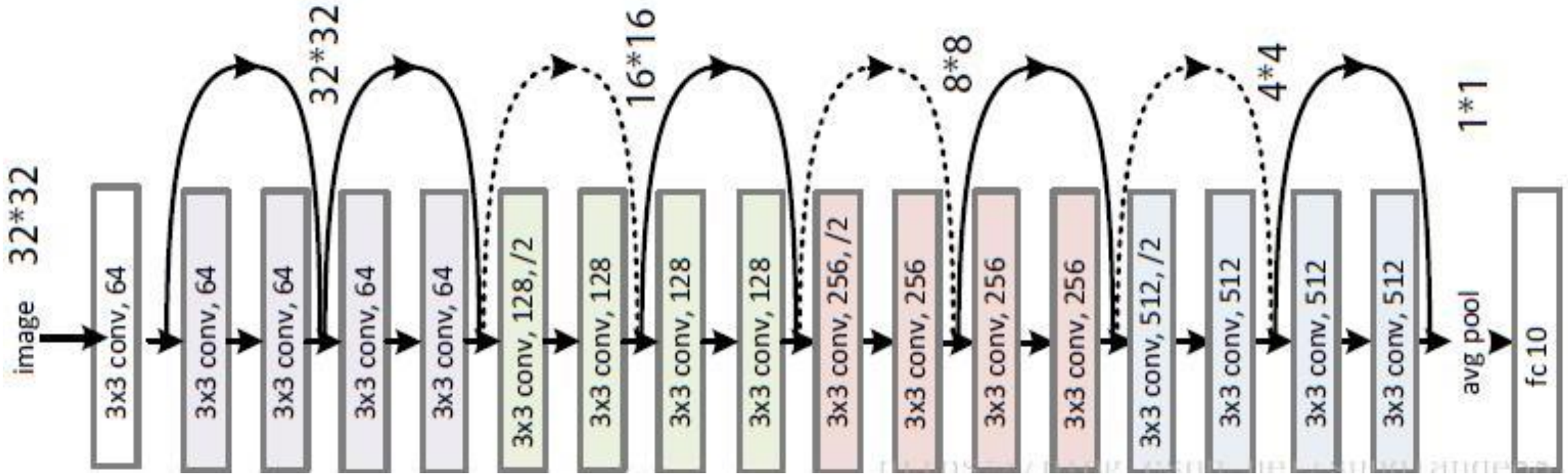
«сжимающие» и «восстанавливающие» свёрточные блоки

Красные слои содержат свёртку, нормализацию по батчу для стабильности обучения и функцию активации ReLU. Она помогает отбросить незначительные признаки и выбрать только нужную информацию. Зелёный слой — пулинг. Здесь изображение сжимается

После того как нейросеть определила, что на изображении, она определяет, где находится этот объект. Для этого она постепенно увеличивает его и применяет повышающую дискретизацию.

Архитектура ResNet

Каждые несколько строк происходит пулинг и сжимает изображение, как в U-Net

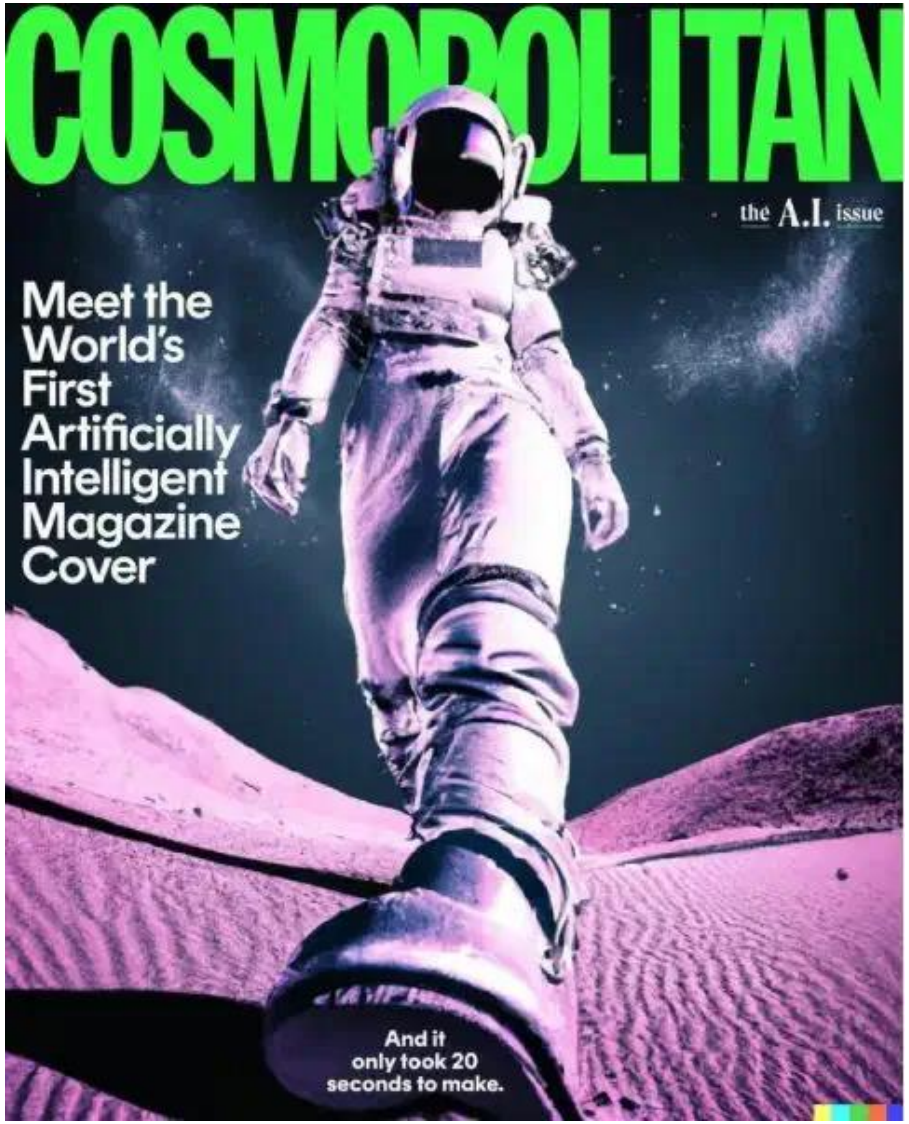


ResNet тоже состоит из нескольких типовых шагов с типовыми операциями: свёртка, нормализация и пулинг. У ResNet есть особенность: после серии свёрточных блоков следуют несколько полносвязных слоёв.
Это стандартный подход для задач классификации изображений.

Генерация изображений

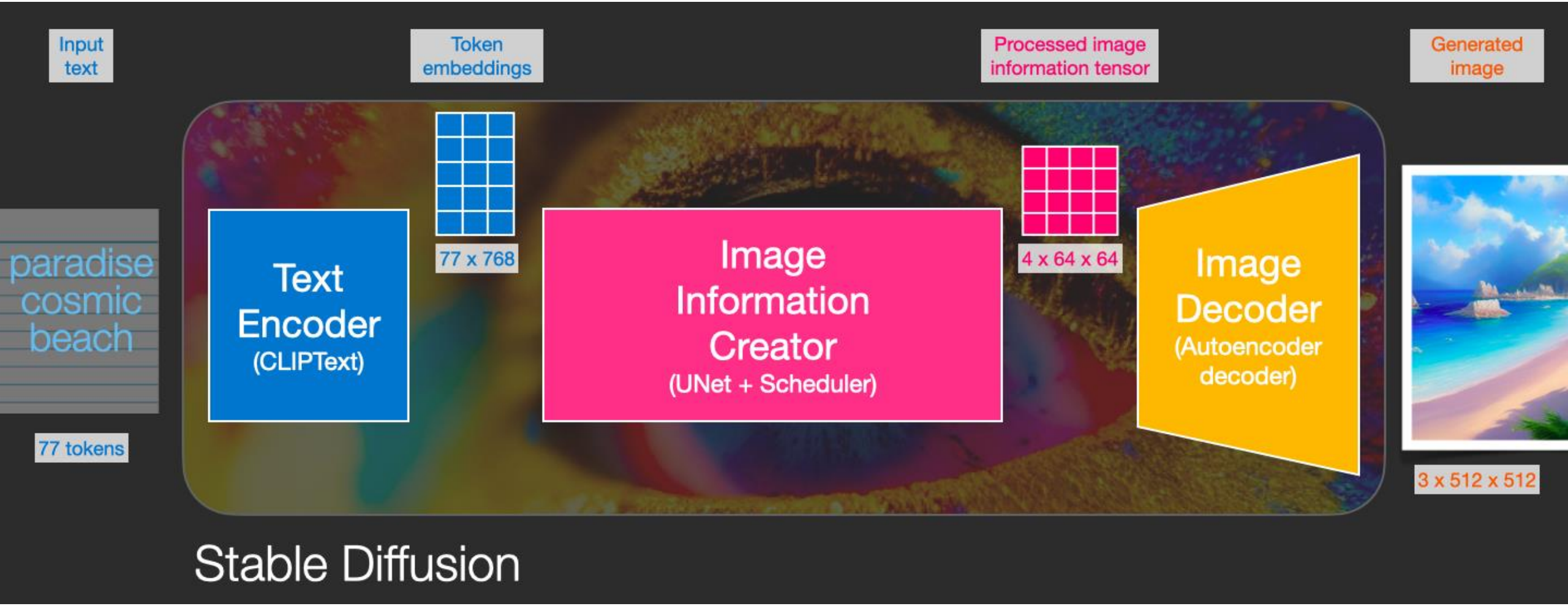
Ещё одно популярное направление, где используют свёрточные нейросети, — генеративное искусство.

Самые популярные модели для генерации изображений
- Midjourney,
- Stable Diffusion,
- Dream,
- DALL-E 2 и
- ruDALL-E.



Обложка Cosmopolitan, нарисованная в компании OpenAI с помощью нейросети DALL-E 2

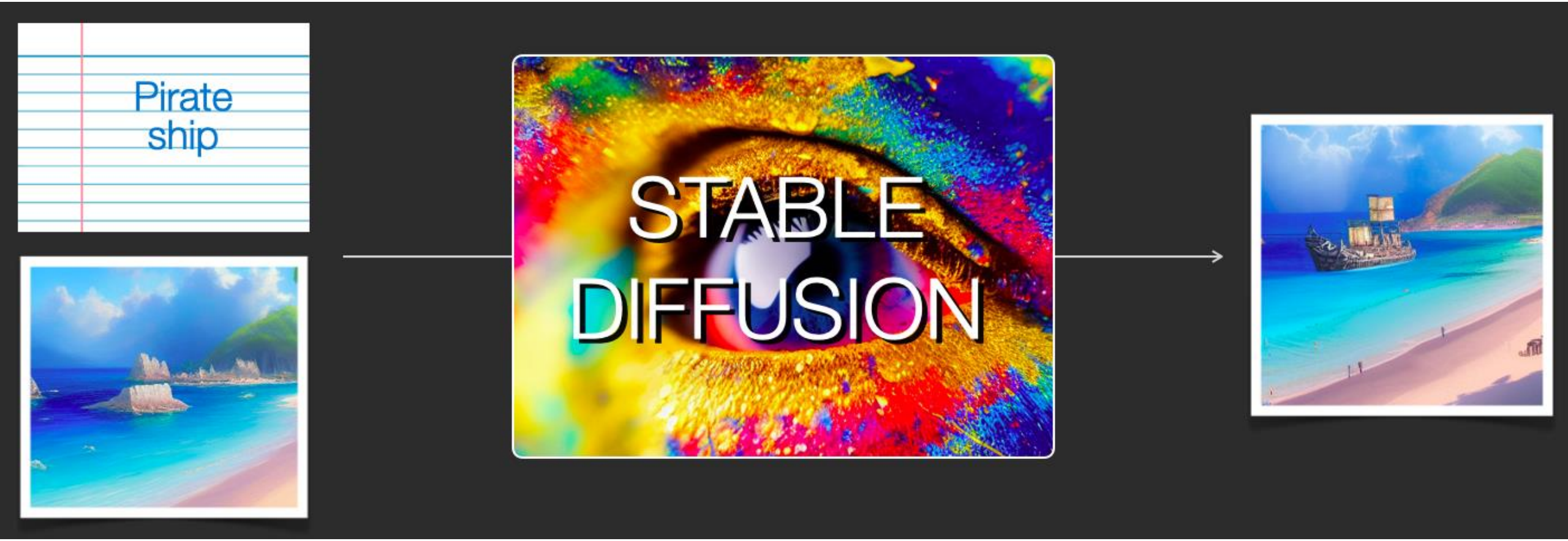
Модель Stable Diffusion – шаг 1



Кодировщик текста — это специальная языковая модель Transformer . Она получает на входе текст и выдаёт на выходе список чисел (вектор), описывающий каждое слово/токен в тексте.

Генератор изображений выполняет два этапа:
Создание информации изображения
Декодер изображений

Stable Diffusion – шаг 2



С какими задачами свёрточные нейросети не справляются

- Свёрточные нейронные сети плохо подходят для анализа глобального контекста, например смысла текстов.
  - В изображении элементы, которые анализирует сеть, находятся рядом.
  - В текстах более длинные связи между элементами: между началом и концом предложения может быть много слов.
  - Такие нейросети не подходят для табличных данных, потому что они, в отличие от пикселей, разнородные: где-то текст, где-то дата, где-то — процент.
- Нейросетям сложно анализировать изображения разного масштаба: например, несколько документов разного формата — A3, A4 и A5 — с одинаковым текстом.

Архитектуры свёрточных нейронных сетей

Они похожи, но используются для разных задач

- классификации
- для сегментации
- детекции
- генерации

Классификация изображений

Задача классификации заключается в том, чтобы для представленного изображения выдать метку, какому классу оно принадлежит

Набор классов может быть заранее определен или быть неизвестным

Классификация или кластеризация

Рассмотрим задачу обучения с учителем, т.е. набор классов заранее определен и есть некий набор данных с истинной разметкой.

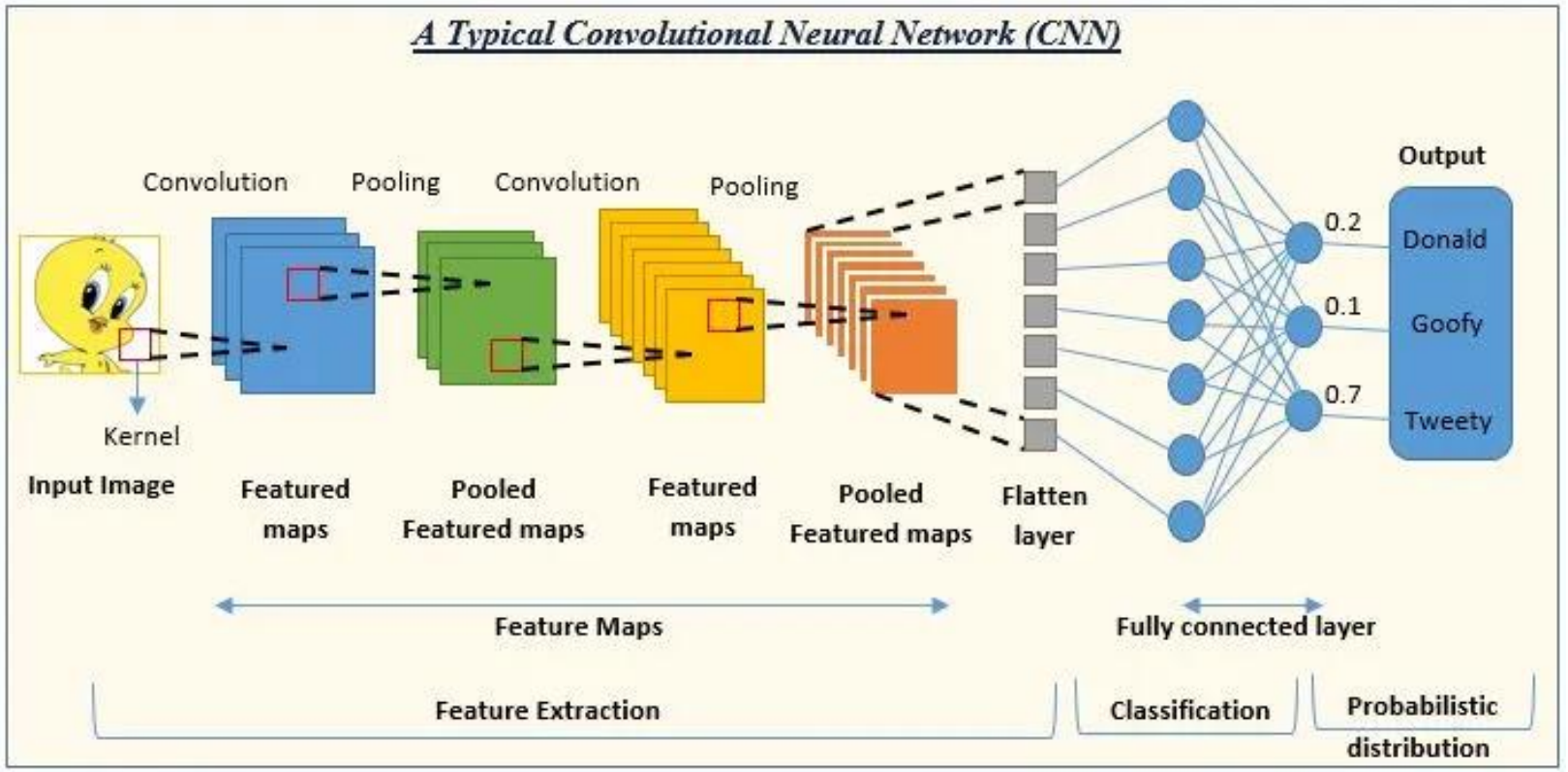
Задача классификации

является самой простой задачей в области компьютерного зрения, для решения которой применяются алгоритмы машинного обучения

большинство исследователей, при разработке новой архитектуры, проверяют её на задачах классификации, а потом адаптируют для других задач

ручная аннотация изображений для данной задачи самая простая и дешевая

Базовая архитектура нейронной сети для решения задачи классификации



В большей степени каждая конкретная архитектура определяет стадию извлечения признаков,

головная часть — обычно один слой или пара полносвязных слоёв

Transfer learning

Очень часто та часть, которая в модели используется для нахождения признаков,
в последующем используется для других задач,
например,
детекции объектов или классификации других классов объектов

Это называется transfer learning
и позволяет воспользоваться знаниями, полученными на задаче с очень большим набором данных,
в другой задаче

Чем трансферное обучение отличается от обычного

Transfer Learning — это просто дообучение ML-алгоритмов для решения других задач

	Классическое машинное обучение (ML)	Трансферное обучение (TL)
Обучение	С нуля	На основе предобученной модели
Вычислительные затраты	Высокие	Низкие
Необходимый объем данных	Большой	Малый
Использование знаний	Каждая модель обучается независимо	Использует знания из предварительно обученной модели
Время достижения оптимальной производительности	Длительное	Быстрое
Эффективность	Менее эффективна при ограниченных ресурсах и данных	Более эффективна при ограниченных ресурсах и данных

AlexNet

Эта архитектура была представлена на конференции NIPS (Neural Information Processing Systems) в 2012 году

Она стала первой нейронной сетью, которая смогла победить на конкурсе классификации изображений

Сама архитектура является простой и практически в точности повторяет ту, которую разработал Я. Лекун в 1998 году.

Ключевыми особенностями работы стали
- сам факт участия в конкурсе и победа в нём,
- и обучение модели с помощью градиентного спуска на видеокарте.

В основе архитектуры AlexNet находятся:

- свёрточный слой: служит для извлечения признаков из переданного изображения или карты признаков предыдущего слоя
- функция активации: некая нелинейная функция, добавляющая нелинейность всей модели
- слой дискредитации: служит для снижения размерности переданной карты признаков, чтобы обеспечить иерархичность признаков и нелинейность
- полносвязный слой: анализирует извлеченные признаки и формирует вектор вероятностей для каждого из классов

AlexNet

Таким образом, классификационная модель возвращает вектор из N элементов, где N - количество поддерживаемых классов.

Каждый элемент - вероятность, что переданное на вход изображение принадлежит этому классу. Сумма всех вероятностей должна равняться единице.

По итогу, качество модели в рамках соревнования на ImageNet 2012 составила:

- Top-1 Error Rate: 37.5%
- Top-5 Error Rate: 17.0%

Показатели точности алгоритма в задаче классификации

Top-1 и Top-5 Error Rate — это показатели точности алгоритма в задаче классификации.

Top-1 Error Rate измеряет, как часто алгоритм не назначает самый высокий балл правильному классу.

Top-5 Error Rate иллюстрирует процент случаев, когда классификатору не удалось включить точный класс в свои Топ-5 прогнозов.