VLMs

Рекомендуемые источники

1. помимо линкнутых статей,
2. https://nanonets.com/blog/bridging-images-and-text-a-survey-of-vlms
3. Обзор: “A Survey of Vision-Language Pre-Trained Models” [IJCAI’22]
4. [2023-2024] Vision-Language Models for Vision Tasks: A Survey arxiv

Специализированные модели

Ранее - специализированные модели:

- решают одну задачу
- под каждую надо собирать свой размеченный датасет

Базовые модели

Базовые модели (**foundation models**):

- способны решать различные задачи
- легко адаптируются под новые задачи (0-shot, few-shot)

В CV:

- универсальные энкодеры
  - CLIP
- VLMs в узком смысле
  - LLM с процессингом изображений (LLaVa, Flamingo)
- 0-shot, open-world CV-таски
  - сегментация (SAM)
  - детекция (Owl-ViT, Grounding-DINO, YOLO-World)

VLM duality

2 значения VLM:

1. широкое: модель, которая процессит 2 модальности (текст и картинки)
2. узкое: расширение LLM, чтобы она на вход (а иногда и выход) могла выдавать картинки

VLM архитектуры

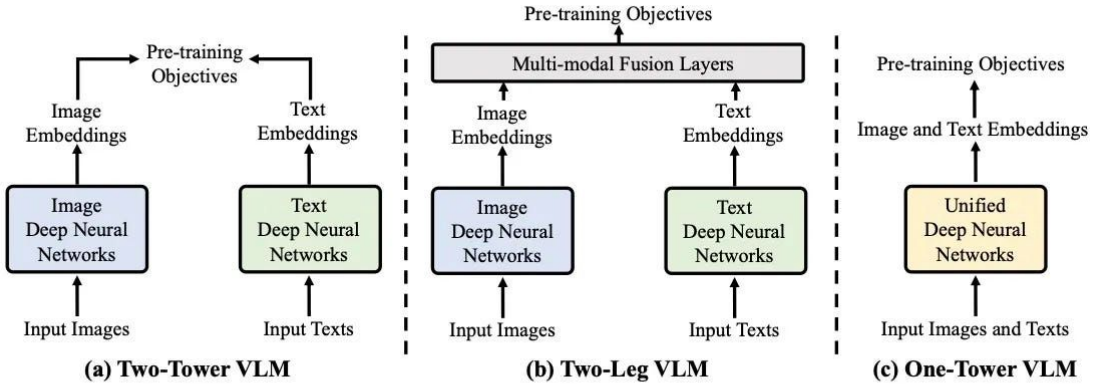


Fig. 5: Illustration of typical VLM pre-training frameworks.

source: https://arxiv.org/pdf/2304.00685

По смешению представлений:

a) late fusion
b) deep (=mid-level) fusion
c) early (=shallow) fusion

По годам:

a) 2021
b) 2022-2023
c) 2023+

VLM обучение

Training objectives:

- contrastive loss - alignment соотвествующих текстовых и картиночных представлений
- generative loss - VLM учат генерить текст на основе картинок и текста
  - часто генерит спец токены ббоксов и прочего
  - иногда также генерит картинки

VLM обучение

Возможные шаги:

1. предобучение img-энкодера и txt-энкодера / LLM по отдельности
2. fine-tuning: обучение с разморозкой разных компонент
3. instruction-tuning: спец. дообучение модели чатиться

Датасеты

a list of: link

- LAION-5B (2022): 5B, publicly available, contains web-scraped image-text pairs
- WIT = Wikipedia-based Image Text (2021): 37M, encyclopedic knowledge
- CommonPool (2023): 12B, by Laion link

Подходы к данным

2 опции:

1. GPT-like: собрать очень много (~6B*) неочищенных данных из интернета

2. Molmo-like: собрать много (~700k) очищенных

* 6B - объем датасета для обучения LLAMA 3.1V

Бенчмарки

HF leaderboard:
https://huggingface.co/spaces/opencompass/open_vlm_leaderboard

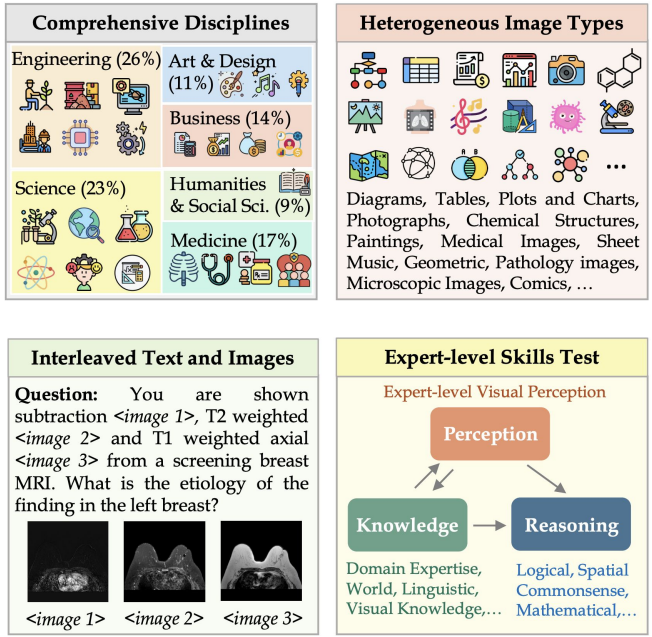
На основе https://github.com/open-compass/VLMEvalKit

- MMMU
- MME
- AI2D
- …

MMMU

https://mmmu-benchmark.github.io/

11k вопросов+картинок



Accounting: Financial Accounting

Question: From the following Company Y adjusted trial balance, what is the retained earnings to be reported?<image 1>
Option:
(A) $17,800
(B) $16,800
(C) $15,550
(D) $12,280

Adjusted Trial Balance

	Debit	Credit
Cash	$ 32,000	
Accounts receivable	17,300	
Prepaid insurance	6,400	
Land	10,000	
Accounts payable		$ 10,900
Salaries payable		6,000
Common stock		31,000
Retained earnings		4,200
Dividends	8,000	
Service revenue		74,000
Insurance expense	5,600	
Salaries expense	24,000	
Miscellaneous expense	22,800	
	126,100	126,100

<image 1>

GPT-4V: To calculate the retained earnings to be reported for Company Y from the adjusted trial balance, we need to consider the following components:
1. Beginning Retained Earnings: This would be the retained earnings balance at the start of the period. However, since it is not provided, we can assume it is included in the current retained earnings figure in the trial balance or that this is the first year of operation, and as such, it would be zero.
2. Add: Net Income (or Less: Net Loss): Net income is calculated as revenues minus expenses. If expenses exceed revenues, it results in a net loss.
3. Less: Dividends: These are distributions to shareholders, which reduce retained earnings.
From the adjusted trial balance provided, the calculation would be as follows:
Net Income: Service Revenue ($74,000) - (Insurance Expense $5,600 + Salaries Expense $24,000 + Miscellaneous Expense $22,800) = $74,000 - $52,400 = $21,600.
Less Dividends: $8,000. Retained Earnings to be Reported = Beginning Retained Earnings (if not given, assumed to be included in the trial balance) + Net Income - Dividends. The trial balance shows Retained Earnings as $6,000. Assuming this is the beginning balance and adding the net income less dividends: Retained Earnings to be Reported = $6,000 + $21,600 - $8,000 = $19,600. There is no answer matching the option.

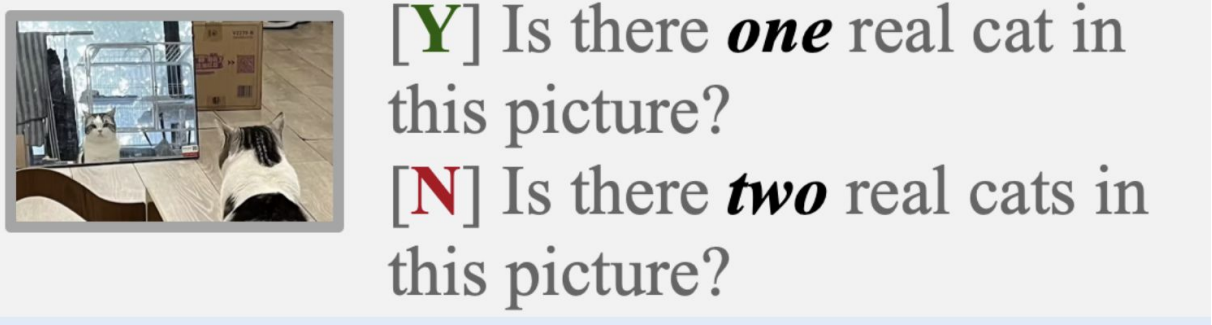
Error Category: Perceptual Error
Error Reason: GPT-4V understands the question and can know how to calculate the Retained Earnings to be Reported. Most of the calculation process is correct. However, in the final steps, GPT-4V wrongly extracts the number of Retained Earnings from the table image (should be $4,200 instead of $6,000).

Ground Truth: (A) $17,800

MME

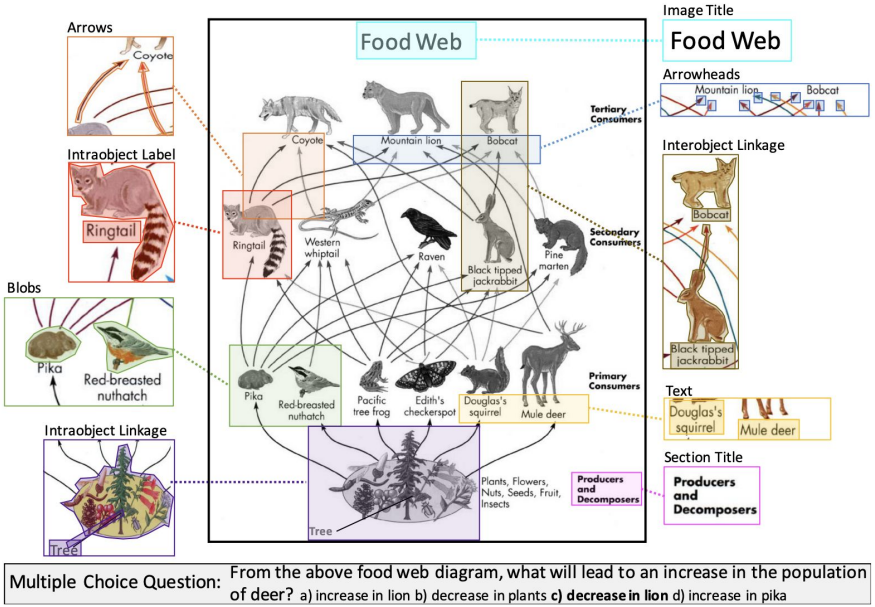
https://arxiv.org/pdf/2306.13394

1k private data: нет в открытом доступе -> модели на них точно не обучались



AI2D

вопросы по (научным) диаграммам

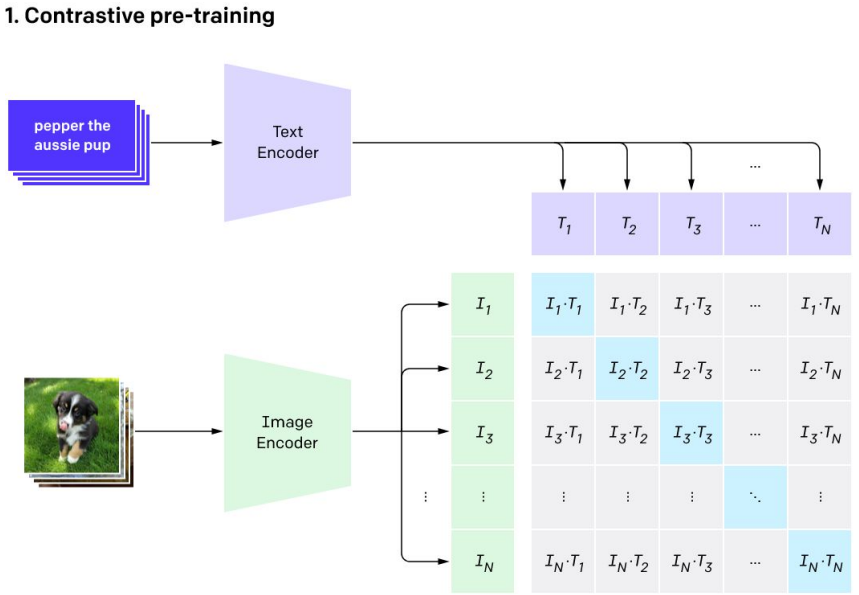


Модели

CLIP: обучение

paper [2021]

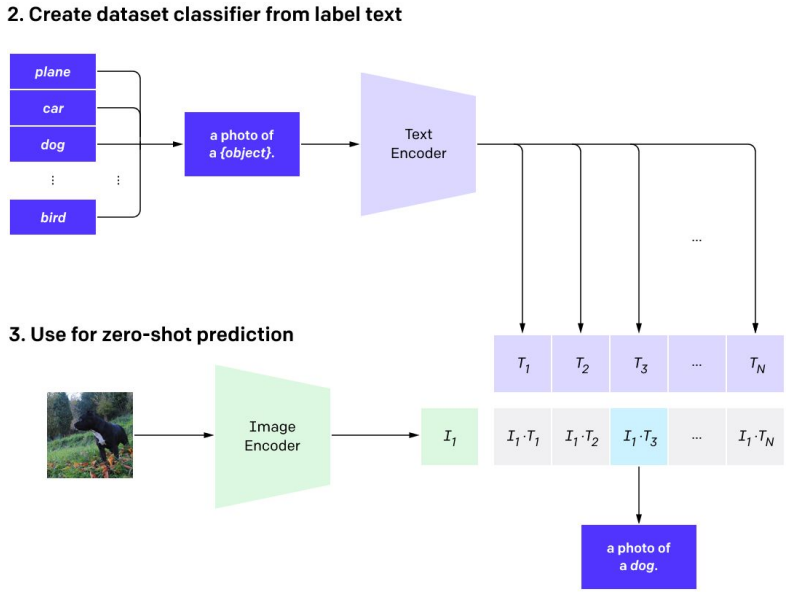
- универсальный энкодер
  - image: ResNet or ViT
  - text: transformer
- 2-tower
- контрастивное обучение: сопоставление эмбеддингов текста и картинок
- много данных:
  датасет WebImageText
  400M пар <текст,картинка>



CLIP: 0-shot классификация

алгоритм:

1. строим эмбеддинги названий классов
2. сравниваем с эмбеддингом изображений
3. выбираем лучший матч
4. предсказываем его класс



VLMs

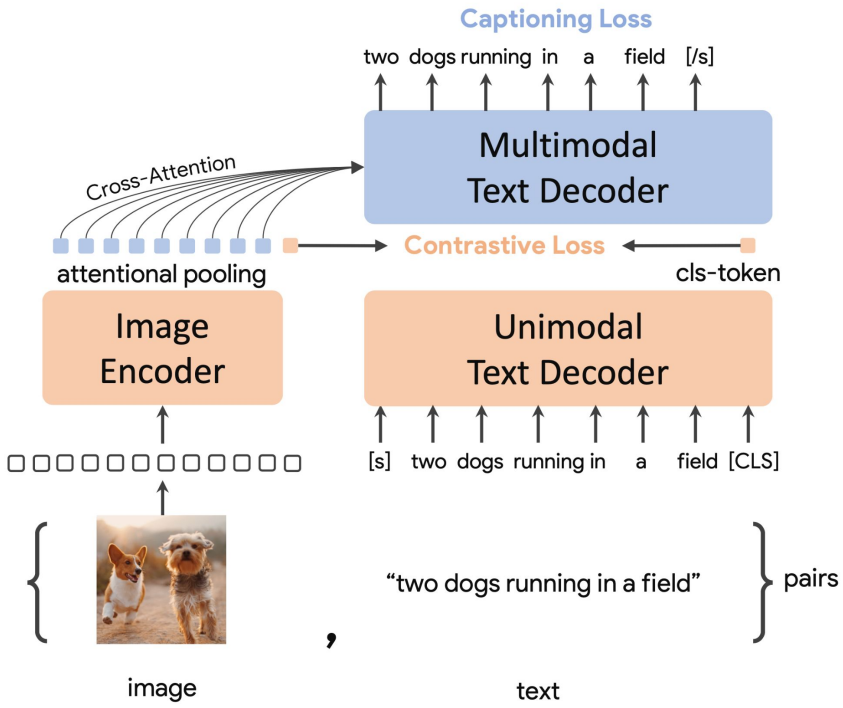
VLM (в узком смысле) - LLM, способная обрабатывать изображения.

Базовая идея: преобразовать изображение в токены, доступные LLM.

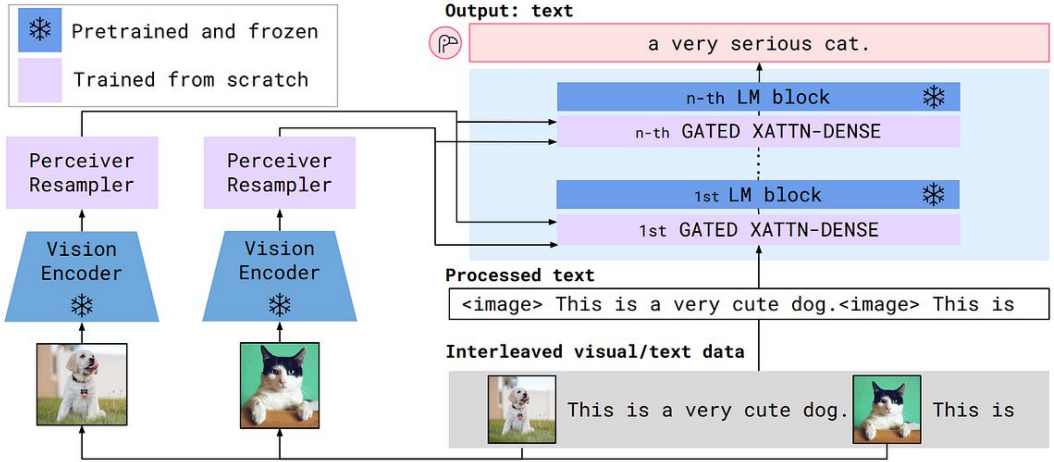
CoCa

[2022] [paper]

- proper VLM
- 2-leg
- generative task



Flamingo



[2022] [paper]

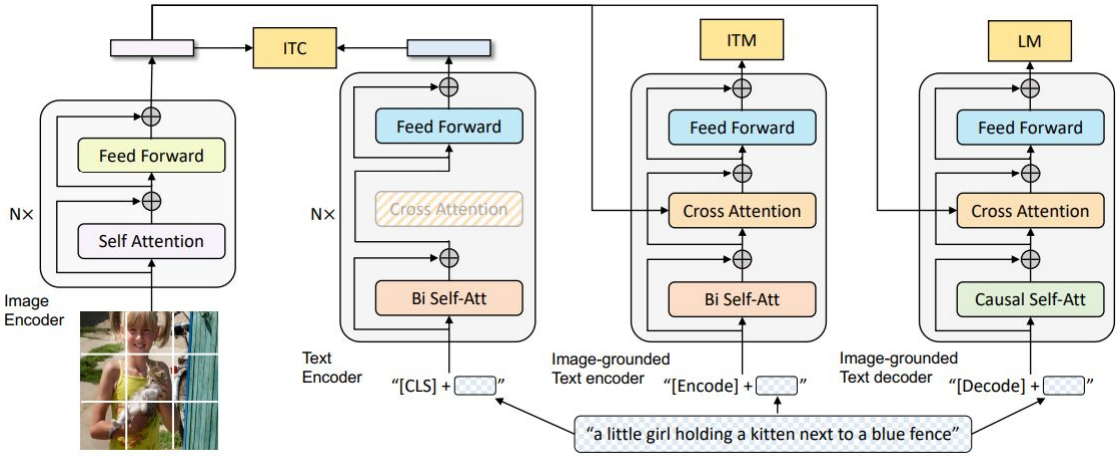
Архитектура: 2-leg

Размер: 3B, 9B, 80B

Обучение:

1. LLM (Chinchilla) - замораживается
2. Vision Encoder предобучается контрастивно вместе с текстом (CLIP-like), затем замораживается
3. обучаются Perceiver Resampler и gated cross-attention layers (language modelling task на мультимодальных web-crawled данных)

BLIP



[2022] paper

Архитекрута: 2-leg

Размер: 200M

Обучение: end-to-end

1. унимодальные энкодеры - image-text contrastive loss (ITC)
2. image-grounded text encoder - image-text matching loss (ITM)
3. image-grounded text decoder - language modeling loss (LM)

Note: ITM - бинарное предсказание матчится ли пара <изображение, текст>

Data: разные датасеты, всего ~10M (~100M)

PALI

[2022] paper

Архитектура:
1-leg (visual tokens подаются в LLMку)

Размер: 17B

Обучение:

1. предобучение
   a. ViT на JFT-3B (image-only) на задаче image classification
   b. mT5 text encoder-decoder
2. тьюнинг:
   a. заморозили ViT & обучают mT5 на WebLI
   b. опциональный full tuning

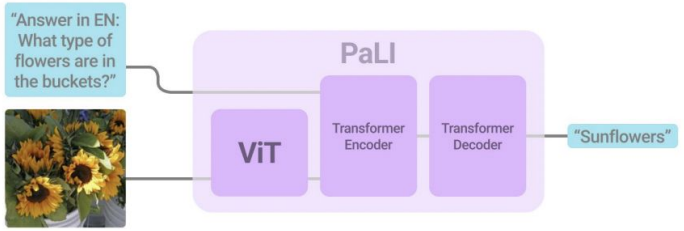
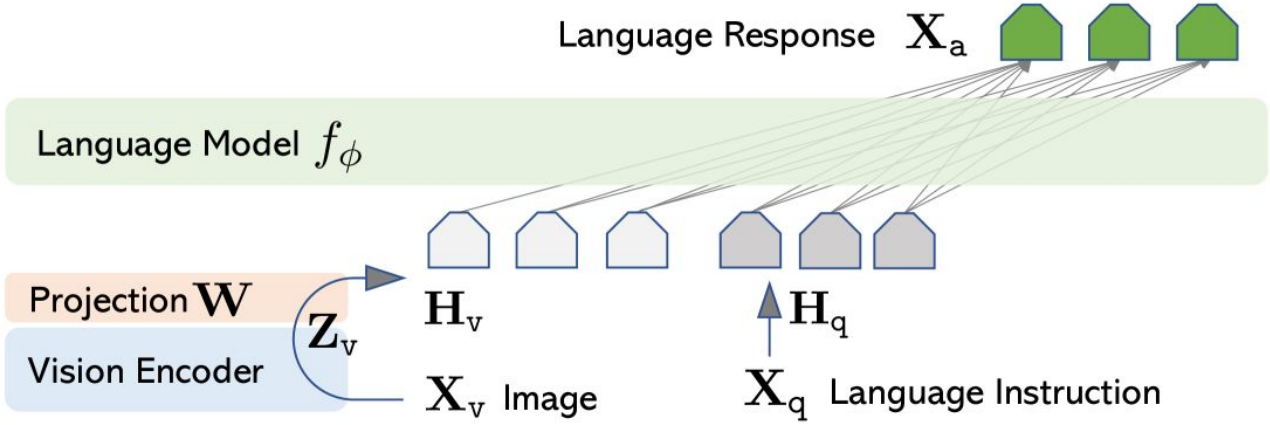


Figure 2: The PaLI main architecture is simple and scalable. It uses an encoder-decoder Transformer model, with a large-capacity ViT component for image processing.

WebLI - закрытый датасет гугла:

- 10B данных
- многоязычный
- на публичных данных
- разнообразные vision-language задачи (VQA, captioning, OCR, …)

LLaVa



[2023] paper - **chat model**

Архитектура:
1-tower (visual tokens подаются в LLMку)

Размер: 7B, 13B

Обучение:

1. initial: берут предобученный vision encoder и LLM (Vicuna)
2. предобучение: замораживают vision encoder и LLM (Vicuna) & обучают только projection
3. тьюнинг: замораживают vision encoder & тьюнят projection & LLM

GPT-4v

[2023]

https://openai.com/index/gpt-4v-system-card/

- вероятно 1-tower
- вероятно большие данные + RLHF
- стартовая точка

Что ещё?..

Открытые:

- Qwen-VL
- Gemma-3 (Google)
- Kosmos-2.5/3 (Microsoft)
- …

Закртые:

- Gemini (Google)
- GPT (OpenAI)
- Claude (Anthropic)
- …

Compression

Model Compression

Опр. **Сжатие моделей (model compression)** - уменьшение размера модели (с сохранением поведения / минимизацией потери качества).

Цели:

- экономия ресурсов для хранения модели;
- экономия вычислительных ресурсов для инференса (RAM, VRAM, energy);
- ускорение инференса/обучения (inference time, latency, throughput)
- сохранение поведения модели / минимизация потери качества.

Подходы к сжатию

Основной подход: quantization

Другие:

- knowledge distillation
- pruning
- *low-rank factorization*
- *NAS*

Метрики

- Метрики вычислительной эффективности
  - FLOPs (операции с плавающей запятой)
  - MACs (операции умножения-накопления) [1 MAC ≈ 2 FLOPs (*, +)]
- Скорость инференса
  - Задержка (latency)
  - Пропускная способность (throughput)
- Потребление памяти
  - Размер модели (количество параметров)
  - Память для промежуточных активаций
- Потребление энергии
- Метрики оценки качества модели
- Робастность

Постановка задачи

Постановки задачи построения сжатой модели:

- **дана большая модель, надо ее сжать**
- строим одновременно замечательную хорошую модель и ее уменьшенные копии
- строим маленькую модель, используем построение большой модели как вспомогательное средство

Quantization

Quantization

Опр. **Квантизация (quantization)** - понижение численной точности (reduction of numerical precision) параметров/активаций.

+ меньше памяти
+ более дешевые (=быстрые) операции
- ошибка квантизации (quantization error) при конвертации
- менее точные вычисления

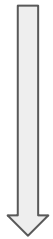
Числовые типы данных

Типы данных:

- целочисленные (integers)
- с фиксированной точкой (fixed point)
- с плавающей точкой (floating point)

В квантизации используют:

- FP32
- FP16, BF16
- INT8
- INT4



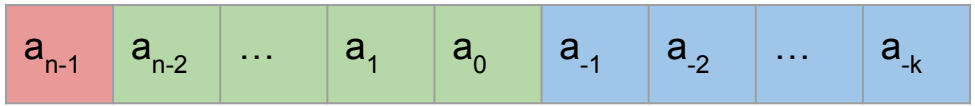
Integers

- unsigned n-bit integer
  - $a = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \ldots + a_1 2^1 + a_0 2^0$
  - диапазон: $0 .. 2^n-1$
- signed n-bit integer
  - $a = - a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \ldots + a_1 2^1 + a_0 2^0$
  - диапазон: $-2^{n-1} .. 2^{n-1}-1$

Fixed-point

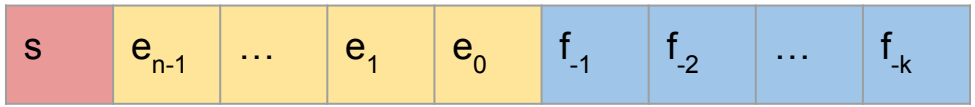
$$a = -a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \ldots + a_1 2^1 + a_0 2^0 + a_{-1}2^{-1} + a_{-2}2^{-2} + \ldots + a_{-k}2^{-k}$$

- знак: 1 bit
- целая часть: n-bit
- дробная часть: k-bit



Floating-point

- знак (sign) s: 1 bit
- порядок (exponent) E: n-bit -> range
- мантисса (fraction) F: k-bit -> precision



(Sub)Normal Numbers

normal numbers ($E \neq 0$)

$a = (-1)^s \times (1 + F) \times 2^{E-bias}$

$F = f_{-1}2^{-1} + f_{-2}2^{-2} + \ldots + f_{-k}2^{-k}$

$E = e_{n-1}2^{n-1} + e_{n-2}2^{n-2} + \ldots + e_0 2^0$

subnormal numbers ($E = 0$)

$a = (-1)^s \times F \times 2^{1-bias}$

$F = f_{-1}2^{-1} + f_{-2}2^{-2} + \ldots + f_{-k}2^{-k}$

Floating-point types

Сравнение:

- FP16 - первоначальный вариант
- BF16 - предложен Google в 2017: сохраняет диапазон (относительно FP32)

	name	spec	s (bits)	E (bits)	F (bits)	max
FP64	Double Precision	IEEE-754	1	11	52	$\sim 10^{308}$
FP32	Single Precision	IEEE-754	1	8	23	$\sim 10^{38}$
FP16	Half Precision	IEEE-754	1	5	10	$\sim 10^{4}$
BF16	Brain Float	Google	1	8	7	$\sim 10^{38}$

Железо

Nvidia GPUs поддерживают

- FP16 since Pascal (2016)
- BF16 since Ampere (2020)

Downcasting effects

меньше битов ->

- меньше памяти
- дешевле операции
- меньше диапазон и/или точность

Отсюда эффекты квантизации.

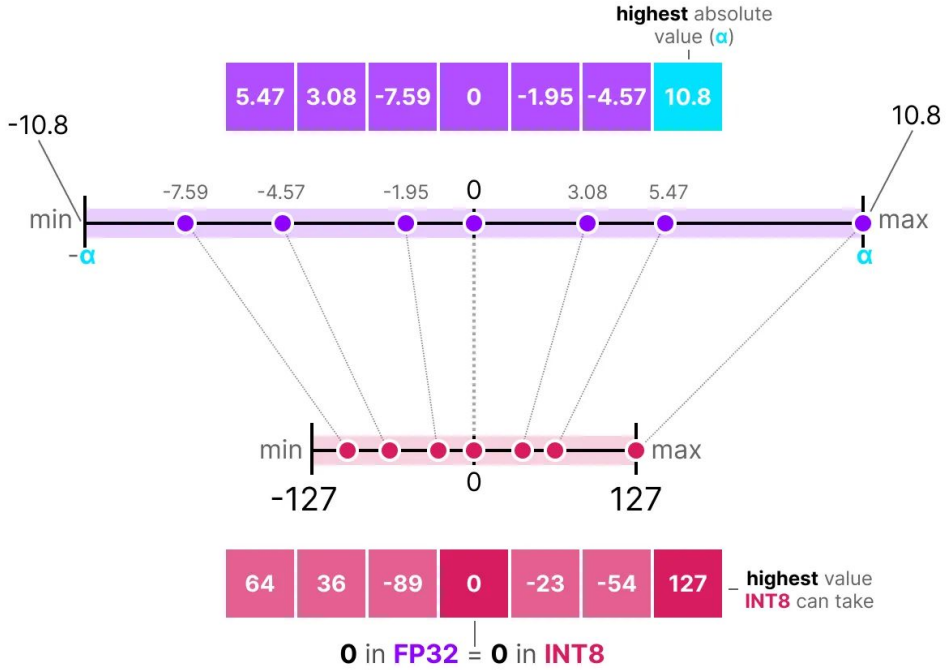
Симметричная линейная квантизация

Основное свойство: 0 -> 0.

Формула: R = sQ, где
R - исходный тензор (fp32)
Q - квантованный тензор (int8)
s - scaling factor (fp32)

Обратное преобразование:
Q = round(R / s)

S = max(abs(R)) / max(abs(Q))



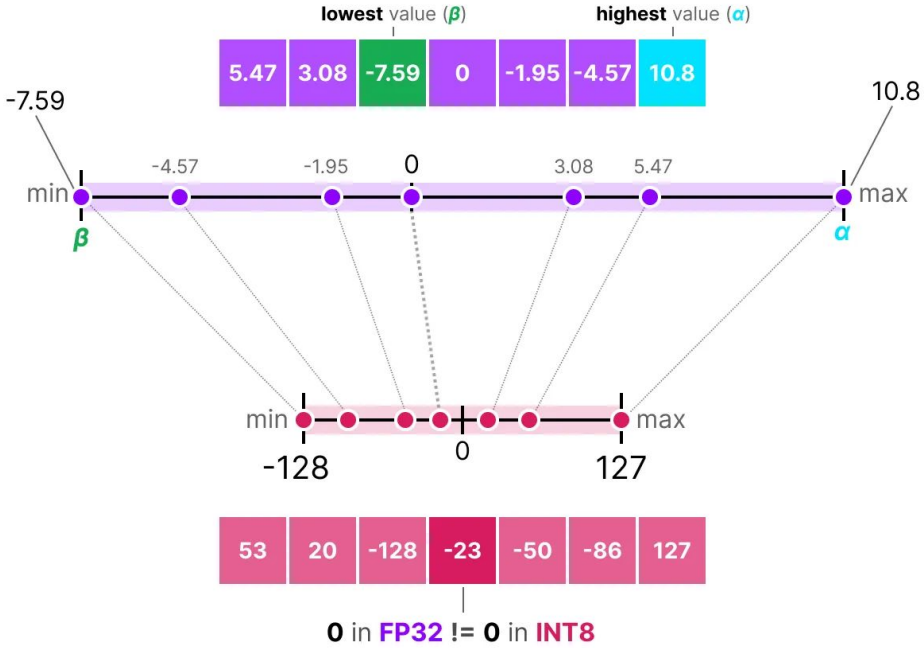
Асимметричная линейная квантизация

Основное свойство: 0 -> z.

Формула: R = s(Q-z), где
R - исходный тензор (fp32)
Q - квантованный тензор (int8)
s - scaling factor (fp32)
z - zero point (int8)

Обратное преобразование:
Q = round(R / s + z)

S = (max(R)-min(R)) / (max(Q)-min(Q))
z = round(min(Q) - min(R) / s)



Матричное умножение

$$Y = WX = s_W(Q_W - z_W)s_X(Q_X - z_X) = s_W s_X(Q_W Q_X - z_W Q_X - z_X Q_W + z_W z_X)$$

Матричное умножение FP32 -> матричное умножение INT8

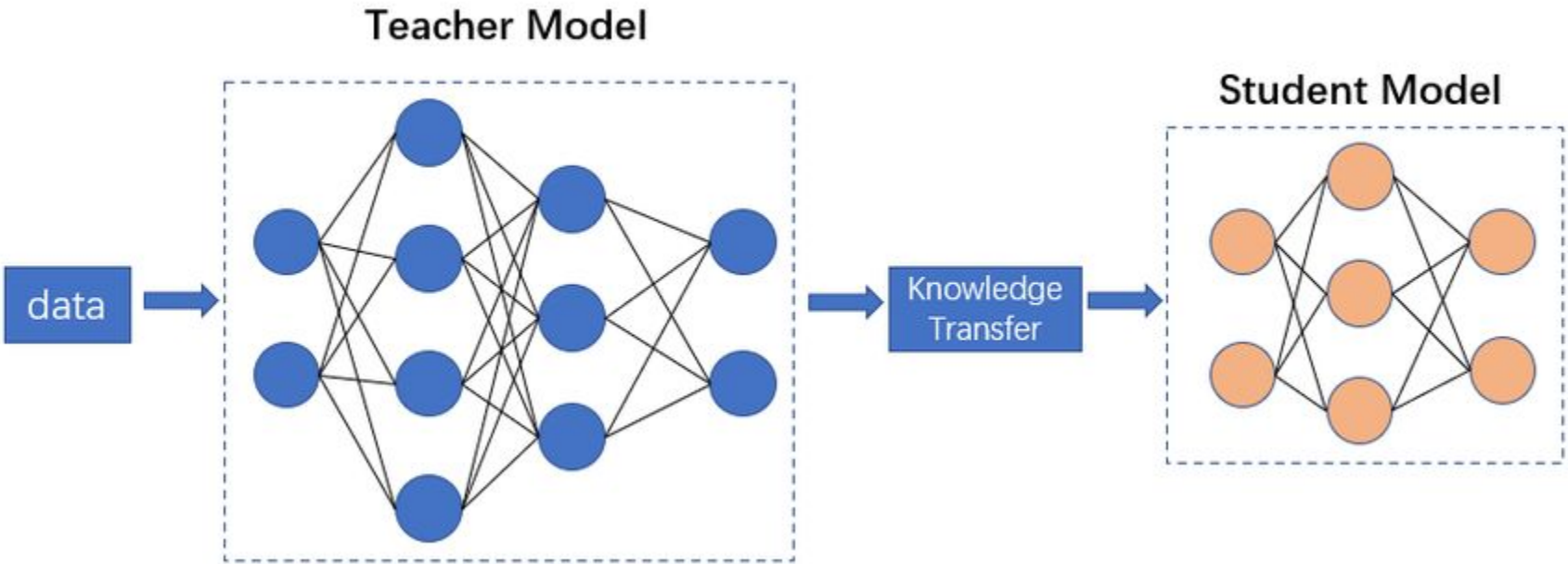
Повышение качества

- Посттренировочная квантизация (PTQ, Post-Training Quantization) - Модель обучается в FP32, затем сжимается.
  - квантизация весов
  - квантизация активаций
- Квантизация во время обучения (QAT, Quantization Aware Training) - Модель обучается с имитацией квантования (fake quantization).
- Mixed-Precision Quantization - Разная квантизация на разных слоях.

Другие виды сжатия

Knowledge Distillation

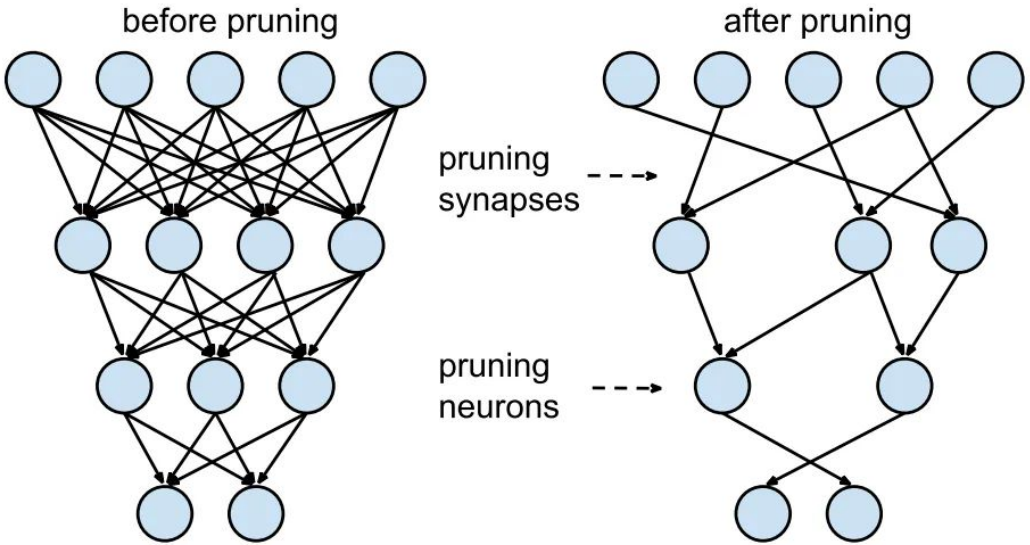
Опр. Knowledge distillation - техника переноса знаний от большей модели (учителя) к меньшей (ученику).



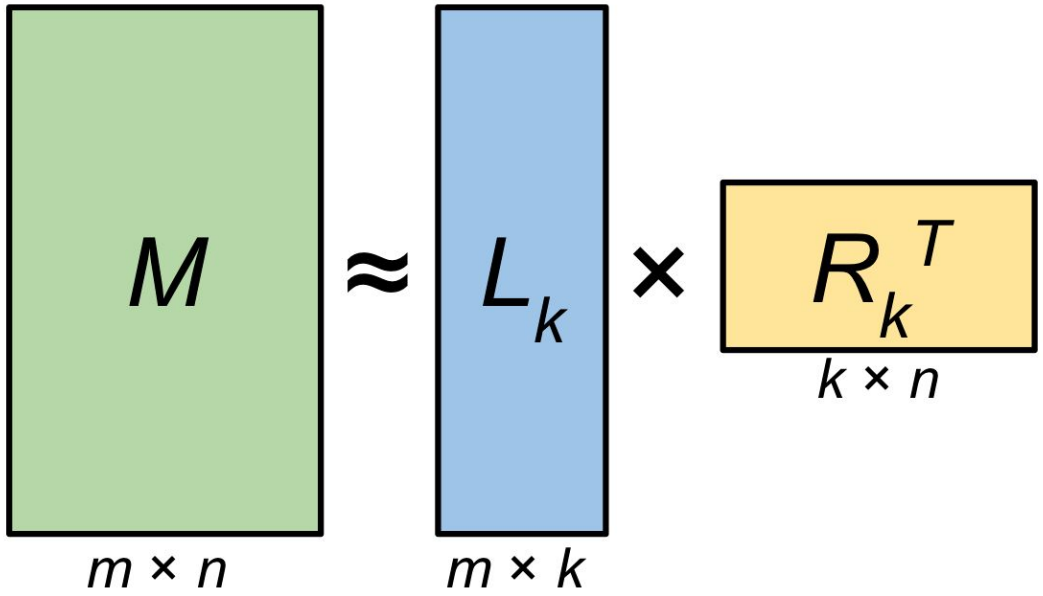
source: https://www.researchgate.net/figure/Principle-of-knowledge-distillation-model_fig2_369046104

Pruning

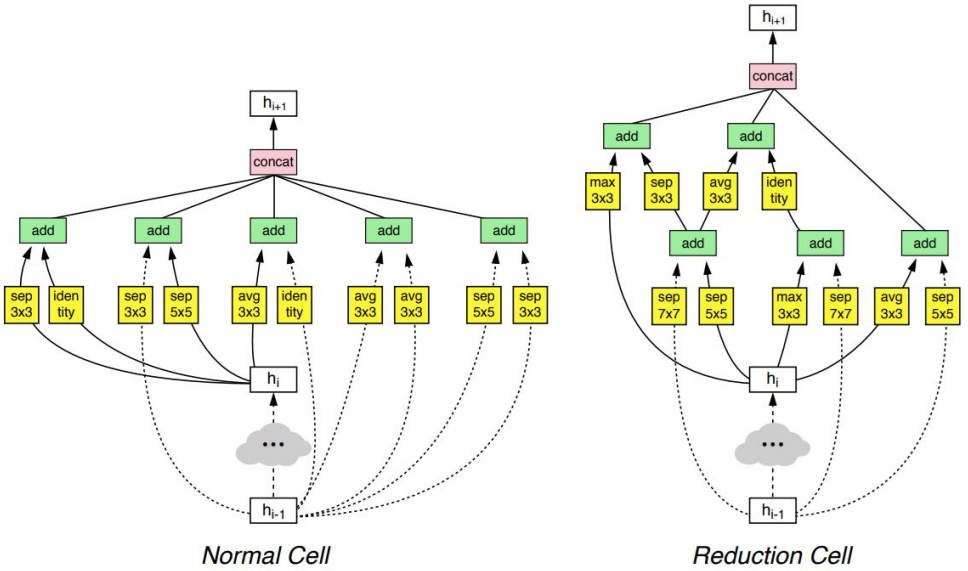
Опр. **Прунинг (pruning)** - удаление нейронов / синапсов.



Low-rank factorization



NAS


h_{i+1}
concat
add
add
add
add
add
sep 3x3
iden tity
sep 3x3
sep 5x5
avg 3x3
iden tity
avg 3x3
avg 3x3
sep 5x5
sep 3x3
h_i
h_{i-1}
Normal Cell
h_{i+1}
concat
add
add
max 3x3
sep 3x3
avg 3x3
iden tity
add
add
add
sep 7x7
sep 5x5
max 3x3
sep 7x7
avg 3x3
sep 5x5
h_i
h_{i-1}
Reduction Cell

Сравнение

Quantization:

+ high compression rate
+ high speed up
+ native hardware support
+ easy to use (implement)
+ moderate performance drop (w/o fine-tuning)
- quality might not be recoverable even with fine-tuning
+ but often is

Рекомендуемые источники

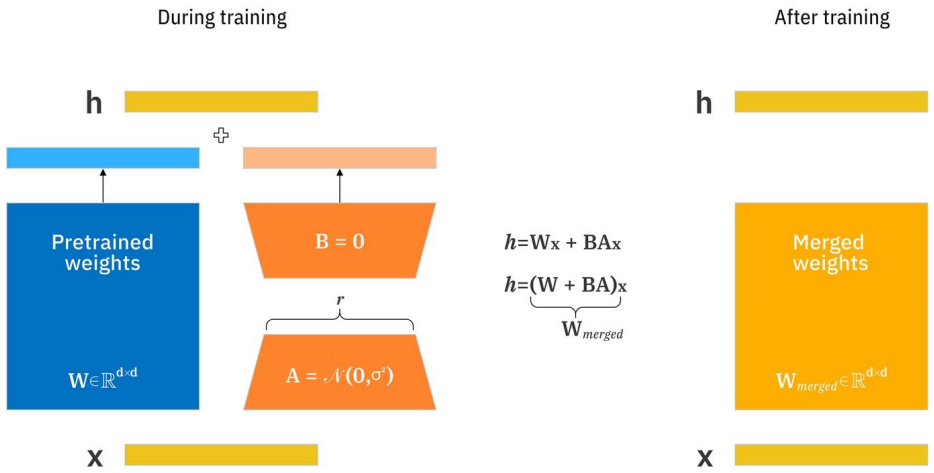
- J. Gou et al., 2021. Knowledge Distillation: A Survey https://arxiv.org/pdf/2006.05525
- H. Cheng et al., 2024. A Survey on Deep Neural Network Pruning-Taxonomy, Comparison, Analysis, and Recommendations https://arxiv.org/abs/2308.06767
- Y. He, L. Xiao, 2023. Structured Pruning for Deep Convolutional Neural Networks: A survey https://arxiv.org/abs/2303.00566
- https://efficientml.ai
- https://learn.deeplearning.ai/courses/quantization-fundamentals
- https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-quantization

Тьюнинг

LoRA

[2021] [paper]

- обучаем не веса, а дифф
- в low rank разложении



QLoRA

QLoRA = LoRA на квантизованной модели

Вопросы?