

Flamingo: a Visual Language Model for Few-Shot Learning

Сизов Михаил. 212

Small Theory Recap

- Few-Shot Learning - модель может дообучаться, если ей подать несколько примеров работы.
- Мульти模альная сеть - работает с разными типами данных на входе (изображения, текст, видео).

Примеры: CLIP и ALIGN. Но они не гибкие, то есть не могут генерировать текст на том же уровне, что и современные LLM. Гибкие существуют (VL-T5) но не умеют в few-shot.

Мотивация и цели






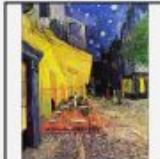



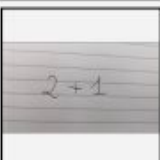

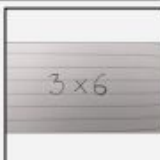
Хотим получить искусственный интеллект, способный обучаться выполнению новой задачи по краткой инструкции (few-shot).

Уже существуют LLM с таким свойством (GPT-3), но мы хотим сделать мультимодальную модель.

Цели:

- Сеть, получающая на вход изображения/видео и текст и выдающая текстовый ответ на запрос.
- Способна к few-shot обучению на мультимодальных данных.

Flamingo - Демонстрация работы

Input Prompt						Completion
	This is a chinchilla. They are mainly found in Chile.		This is a shiba. They are very popular in Japan.		This is	a flamingo. They are found in the Caribbean and South America.
	What is the title of this painting? Answer: The Hallucinogenic Toreador.		Where is this painting displayed? Answer: Louvres Museum, Paris.		What is the name of the city where this was painted? Answer:	Arles.
	Output: "Underground"		Output: "Congress"		Output:	"Soulomes"
	2+1=3		5+6=11			3x6=18

Flamingo - Архитектура

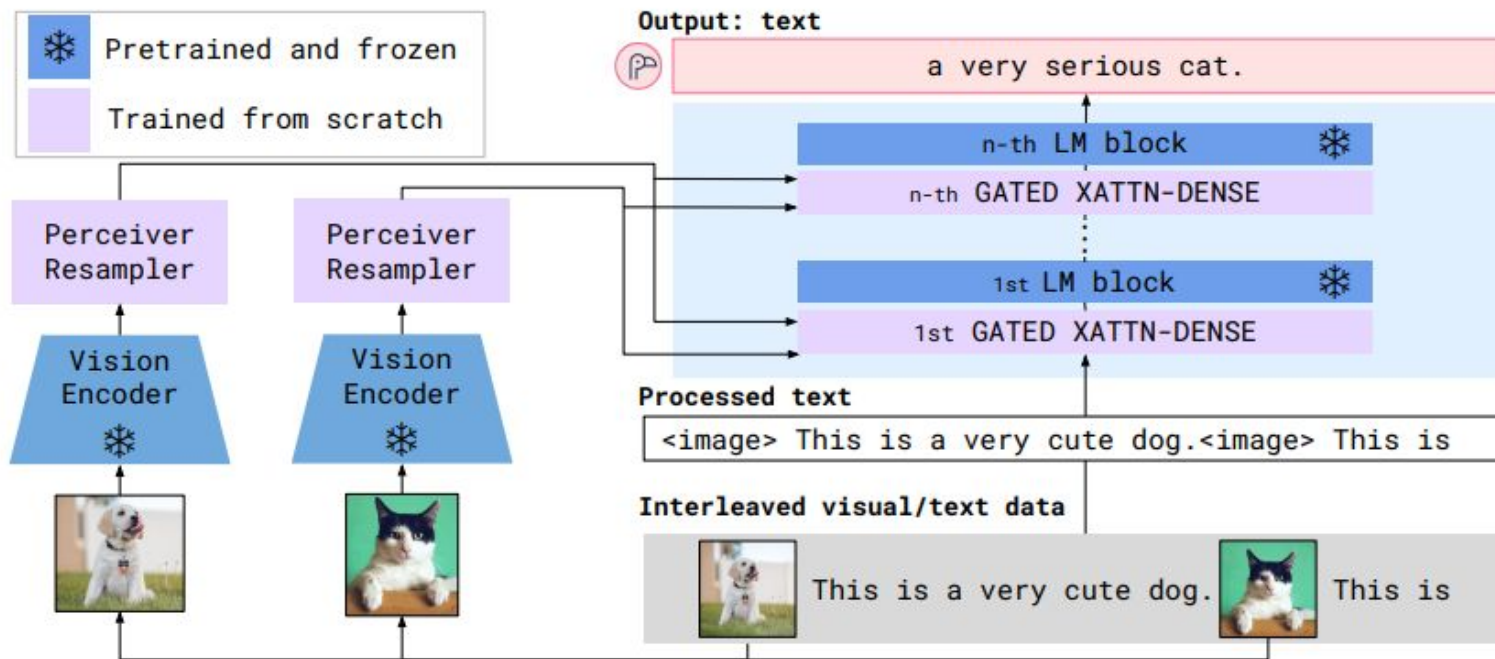


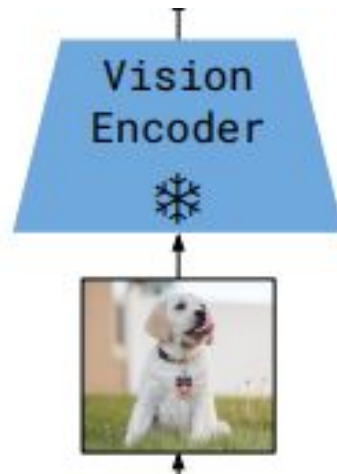
Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

Visual Encoder

Задача: получить признаков из объектов.

Устройство:

- Предобучена и заморожена ❄
- Normalized-Free ResNet
- Архитектура Image Encoder из CLIP
- Обучается на contrastive loss (как CLIP)
- На выходе 2D spatial grid, который с помощью flatten превращается в 1D

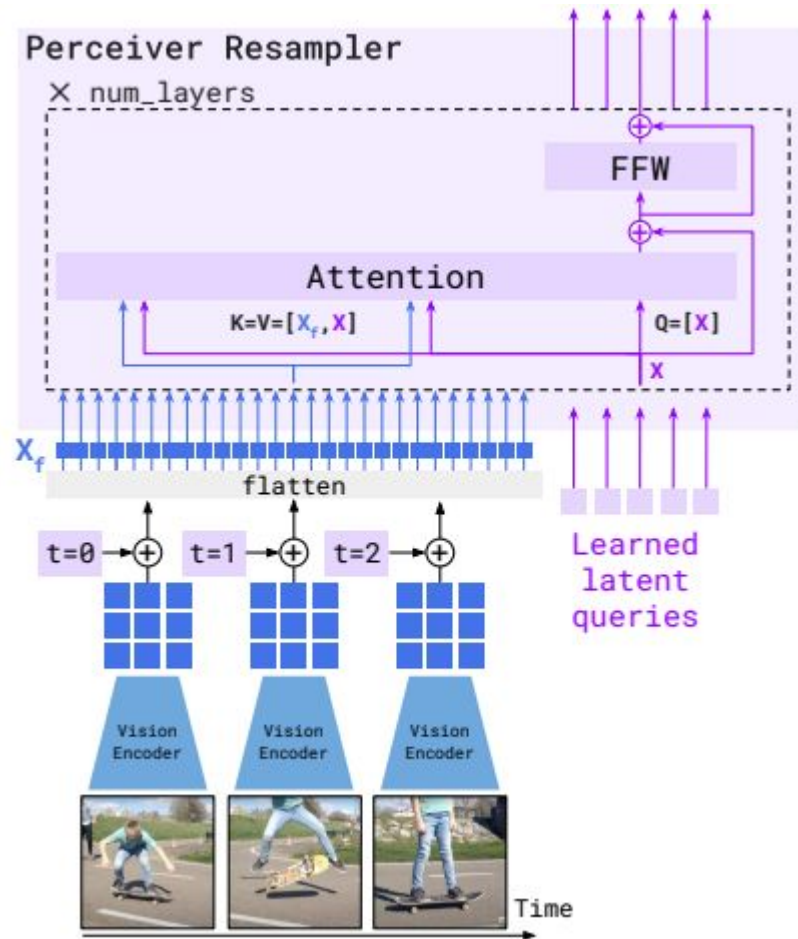


Perceiver Resampler

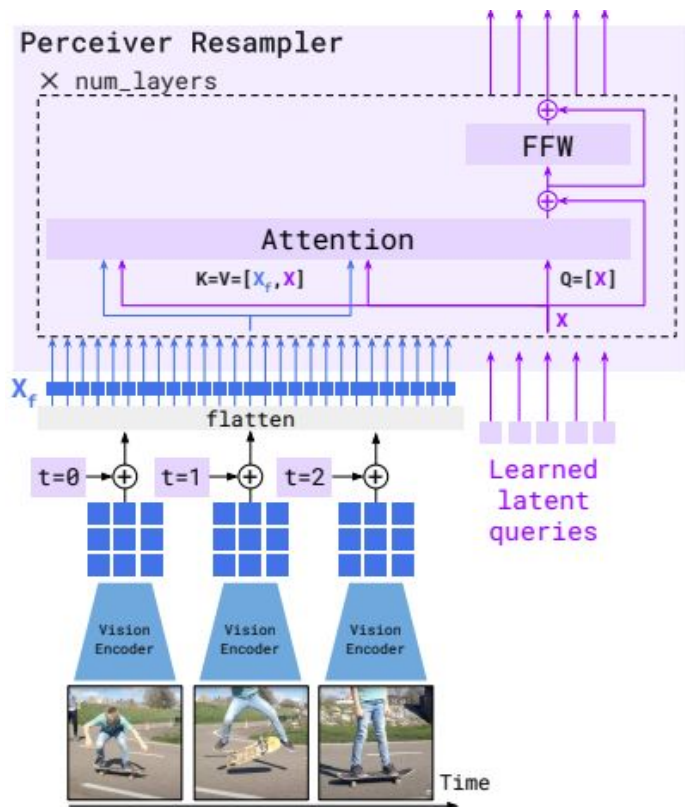
Задача: получение вектора из токенов

Устройство:

- Несколько слоев Attention и Feed Forward
- Q - на первой итерации - обучаемый параметр. На следующих он берется из предыдущей
- K и V - конкатенация вектора признаков и Q
- На выходе вектор из 64 токенов



Perceiver Resampler - Реализация



```
def perceiver_resampler(
    x_f, # The [T, S, d] visual features (T=time, S=space)
    time_embeddings, # The [T, 1, d] time pos embeddings.
    x, # R learned latents of shape [R, d]
    num_layers, # Number of layers
):
    """The Perceiver Resampler model."""

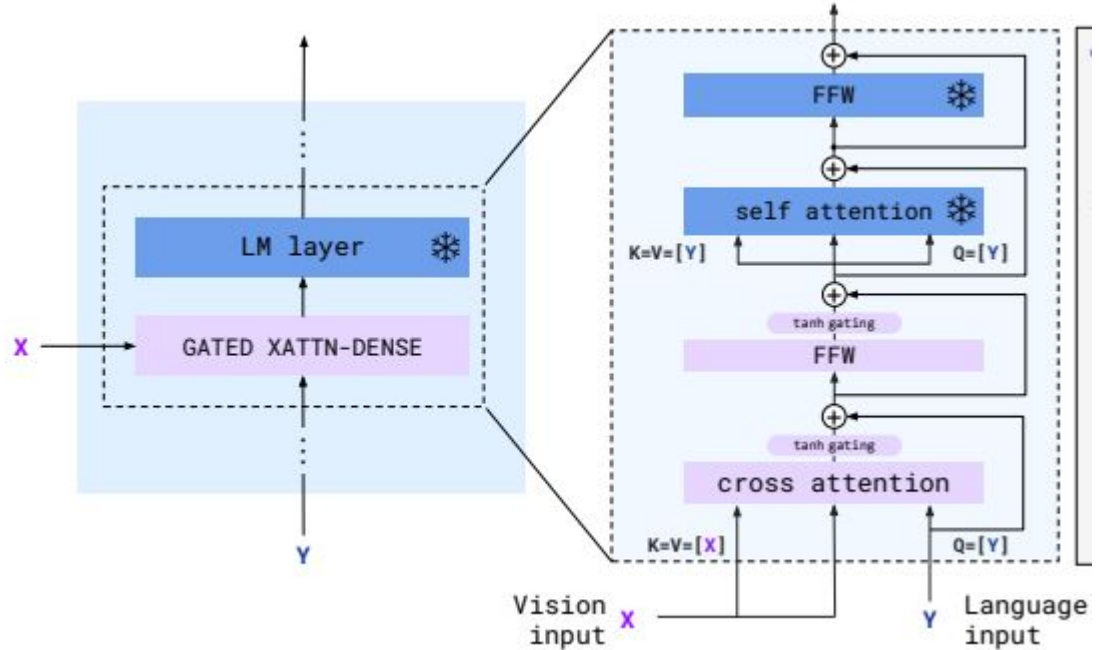
    # Add the time position embeddings and flatten.
    x_f = x_f + time_embeddings
    x_f = flatten(x_f) # [T, S, d] -> [T * S, d]
    # Apply the Perceiver Resampler layers.
    for i in range(num_layers):
        # Attention.
        x = x + attention_i(q=x, kv=concat([x_f, x]))
        # Feed forward.
        x = x + ffw_i(x)
    return x
```


Gated Xattn-dense block

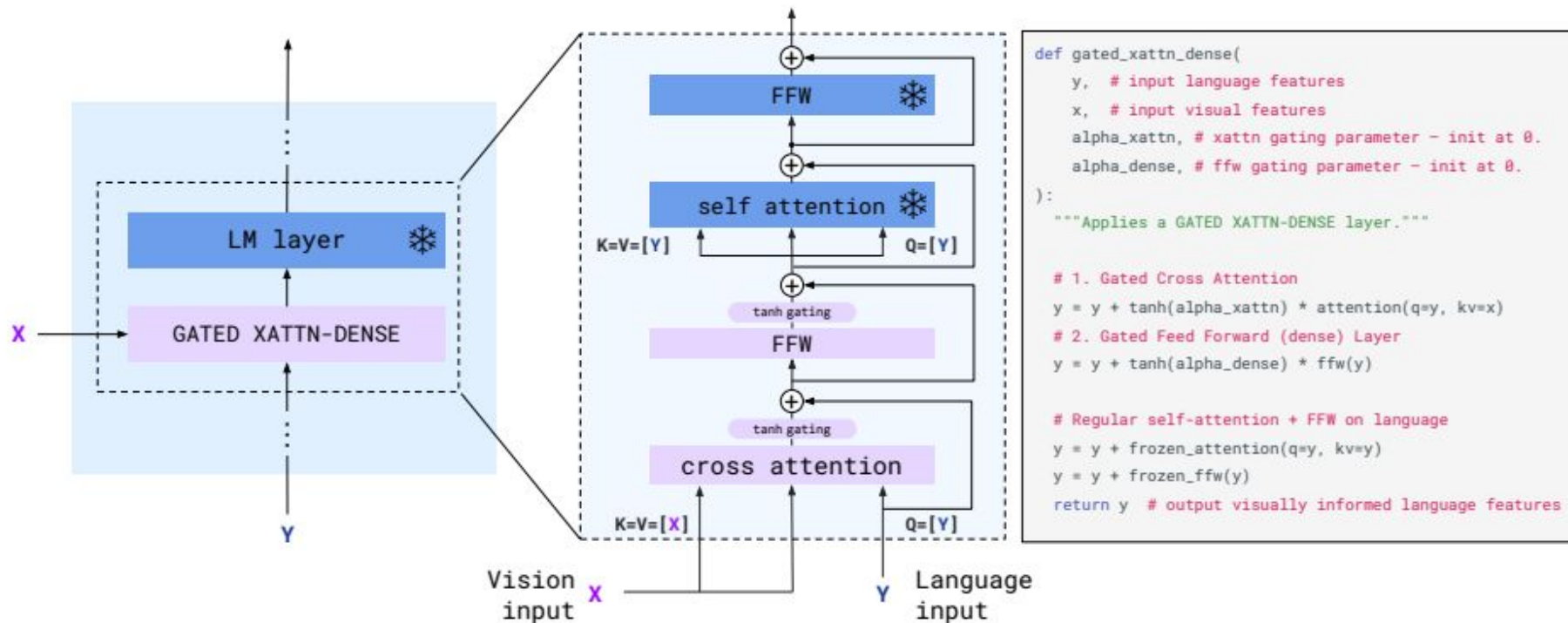
Задача: обучение модели на правдоподобии

Устройство:

- Слои LM заморожены
- Параметр в tanh gating - обучаемый, начальное значение 0
- Модель плавно переходит из языковой в мультимодальную



Gated Xattn-dense block - Реализация



Датасеты

- MultiModal MassiveWeb (M3W) - изображения с текстом из интернета. 256 токенов и изображения, оказавшиеся между этими токенами (до пяти)
- Long Text and Image Pairs (LTIP) - картинки с длинным описанием из датасета ALIGN. Картинки 320 x 320 пикселей, 32/64 токена
- Video and Text Pairs (VTP) - видео с длинным описанием. 320 x 320 пикселей, 32 токена, средняя длительность видео 22 секунды

Обучение

Training objective

Models are trained with a **weighted sum** of dataset specific negative log likelihoods of text (conditioned on visual inputs):

$$\sum_{m=1}^M \lambda_m \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}_m} \left[- \sum_{\ell=1}^L \log p(y_\ell \mid y_{<\ell}, x_{\leq \ell}) \right]$$

\mathcal{D}_m - m_{th} dataset

λ_m - positive scalar weight for the m_{th} dataset

Similar to vision encoder pretraining tuning these weights is important and the **accumulation** strategy for combining data is used.

Оптимизации:

- AdamW
- Linear warmup + flat learning rate
- Subword tokenizer

Результаты - Ablation Study

	Ablated setting	<i>Flamingo</i> 3B original value	Changed value	Param. count ↓	Step time ↓	COCO CIDEr↑	OKVQA top1↑	VQAv2 top1↑	MSVDQA top1↑	VATEX CIDEr↑	Overall score↑
	<i>Flamingo-3B model</i>			3.2B	1.74s	86.5	42.1	55.8	36.3	53.4	70.7
(i)	Training data	All data	w/o Video-Text pairs	3.2B	1.42s	84.2	43.0	53.9	34.5	46.0	67.3
			w/o Image-Text pairs	3.2B	0.95s	66.3	39.2	51.6	32.0	41.6	60.9
			Image-Text pairs→ LAION	3.2B	1.74s	79.5	41.4	53.5	33.9	47.6	66.4
			w/o M3W	3.2B	1.02s	54.1	36.5	52.7	31.4	23.5	53.4
(ii)	Optimisation	Accumulation	Round Robin	3.2B	1.68s	76.1	39.8	52.1	33.2	40.8	62.9
(iii)	Tanh gating	✓	✗	3.2B	1.74s	78.4	40.5	52.9	35.9	47.5	66.5
(iv)	Cross-attention architecture	GATED XATTN-DENSE	VANILLA XATTN	2.4B	1.16s	80.6	41.5	53.4	32.9	50.7	66.9
			GRAFTING	3.3B	1.74s	79.2	36.1	50.8	32.2	47.8	63.1
(v)	Cross-attention frequency	Every	Single in middle	2.0B	0.87s	71.5	38.1	50.2	29.1	42.3	59.8
			Every 4th	2.3B	1.02s	82.3	42.7	55.1	34.6	50.8	68.8
			Every 2nd	2.6B	1.24s	83.7	41.0	55.8	34.5	49.7	68.2
(vi)	Resampler	Perceiver	MLP	3.2B	1.85s	78.6	42.2	54.7	35.2	44.7	66.6
			Transformer	3.2B	1.81s	83.2	41.7	55.6	31.5	48.3	66.7
(vii)	Vision encoder	NFNet-F6	CLIP ViT-L/14	3.1B	1.58s	76.5	41.6	53.4	33.2	44.5	64.9
			NFNet-F0	2.9B	1.45s	73.8	40.5	52.8	31.1	42.9	62.7
(viii)	Freezing LM	✓	✗ (random init)	3.2B	2.42s	74.8	31.5	45.6	26.9	50.1	57.8
			✗ (pretrained)	3.2B	2.42s	81.2	33.7	47.4	31.0	53.9	62.7

Table 3: **Ablation studies.** Each row should be compared to the baseline Flamingo run (top row). Step time measures the time spent to perform gradient updates on all training datasets.

Результаты - Сравнения

Method	FT	Shot	OKVQA (I)	VQAv2 (I)	COCO (I)	MSVDQA (V)	VATEX (V)	VizWiz (I)	Flick30K (I)	MSRVTTQA (V)	iVQA (V)	YouCook2 (V)	STAR (V)	VisDial (I)	TextVQA (I)	NextQA (I)	HatefulMemes (I)	RareAct (V)
Zero/Few shot SOTA	✗		[34] 43.3 (16)	[114] 38.2 (4)	[124] 32.2 (0)	[58] 35.2 (0)	-	-	-	[58] 19.2 (0)	[135] 12.2 (0)	-	[143] 39.4 (0)	[79] 11.6 (0)	-	-	[85] 66.1 (0)	[85] 40.7 (0)
Flamingo-3B	✗	0	41.2	49.2	73.0	27.5	40.1	28.9	60.6	11.0	32.7	55.8	39.6	46.1	30.1	21.3	53.7	58.4
	✗	4	43.3	53.2	85.0	33.0	50.0	34.0	72.0	14.9	35.7	64.6	41.3	47.3	32.7	22.4	53.6	-
	✗	32	45.9	57.1	99.0	42.6	59.2	45.5	71.2	25.6	37.7	76.7	41.6	47.3	30.6	26.1	56.3	-
Flamingo-9B	✗	0	44.7	51.8	79.4	30.2	39.5	28.8	61.5	13.7	35.2	55.0	41.8	48.0	31.8	23.0	57.0	57.9
	✗	4	49.3	56.3	93.1	36.2	51.7	34.9	72.6	18.2	37.7	70.8	42.8	50.4	33.6	24.7	62.7	-
	✗	32	51.0	60.4	106.3	47.2	57.4	44.0	72.8	29.4	40.7	77.3	41.2	50.4	32.6	28.4	63.5	-
Flamingo	✗	0	50.6	56.3	84.3	35.6	46.7	31.6	67.2	17.4	40.7	60.1	39.7	52.0	35.0	26.7	46.4	60.8
	✗	4	57.4	63.1	103.2	41.7	56.0	39.6	75.1	23.9	44.1	74.5	42.4	55.6	36.5	30.8	68.6	-
	✗	32	57.8	67.6	113.8	52.3	65.1	49.8	75.4	31.0	45.3	86.8	42.2	55.6	37.9	33.5	70.0	-
Pretrained FT SOTA	✓		54.4 [34] (10K)	80.2 [140] (444K)	143.3 [124] (500K)	47.9 [28] (27K)	76.3 [153] (500K)	57.2 [65] (20K)	67.4 [150] (30K)	46.8 [51] (130K)	35.4 [135] (6K)	138.7 [132] (10K)	36.7 [128] (46K)	75.2 [79] (123K)	54.7 [137] (20K)	25.2 [129] (38K)	79.1 [62] (9K)	-

Table 1: **Comparison to the state of the art.** A *single* Flamingo model reaches the state of the art on a wide array of image (I) and video (V) understanding tasks with few-shot learning, significantly outperforming previous best zero- and few-shot methods with as few as four examples. More importantly, using only 32 examples and without adapting any model weights, Flamingo *outperforms* the current best methods – fine-tuned on thousands of annotated examples – on seven tasks. Best few-shot numbers are in **bold**, best numbers overall are underlined.

Источники

[Оригинальная статья](#)