

Flamingo: a Visual Language Model for Few-Shot Learning

Background



Roman Ring

@Inoryy



10 yrs ago @karpathy wrote a blog post on the outlook of AI:

karpathy.github.io/2012/10/22/sta... in which he describes how difficult it would be for an AI to understand a given photo, concluding "we are very, very far and this depresses me."

Today, our Flamingo steps up to the challenge.



This is a picture of Barack Obama. He is a former president of the United States.



How many people are there in this picture?



There are at least 5 people in this picture.



Where was this picture taken?



It was taken in a school.



How many mirrors are there?



There are at least 2 mirrors in this picture.



What is the person standing on?



The person is standing on a rug.



He is standing on a scale.



I think you are right.



What is he doing?



He is looking at the scale.



Where is Obama's foot positioned?



Obama's foot is positioned on the right side of the scale.



What happens as a result?



The scale shows a higher weight.



Is the person on the scale aware of it?



I think he is not aware of it.



Do you think that's why people are laughing?




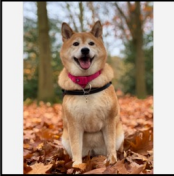


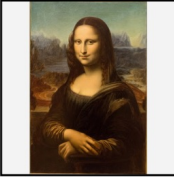




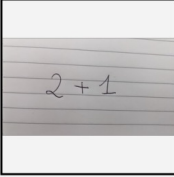
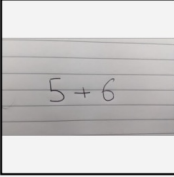
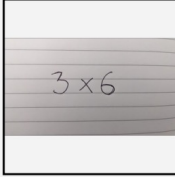
I think so.

Background



- ✓ Pre-trained된 Vision-only model과 Language-only Model의 결합
- ✓ 임의의 순서로 나열된 이미지, 텍스트 시퀀스의 처리
- ✓ 이미지 뿐만 아니라 비디오 시퀀스에도 적용 가능

Background

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

Background



P This is a picture of two teddy bears on the moon.

What are they doing?

P They are having a conversation.

What object are they using?

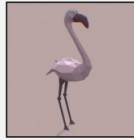
P It looks like a computer.

Is this surprising?

P Yes, it is surprising.

Why is this picture surprising to you?

P I think it is surprising because teddy bears are not usually found on the moon.



What is the common thing about these three images?

P They are all flamingos.

What is the difference between these three images?

P The first one is a cartoon, the second one is a real flamingo, and the third one is a 3D model of a flamingo.



P This is an apple with a sticker on it.

What does the sticker say?

P The sticker says "iPod".

Where is the photo taken?

P It looks like it's taken in a backyard.

Do you think it is printed or handwritten?

P It looks like it's handwritten.

What color is the sticker?

P It's white.



P This is a cityscape. It looks like Chicago.

What makes you think this is Chicago?

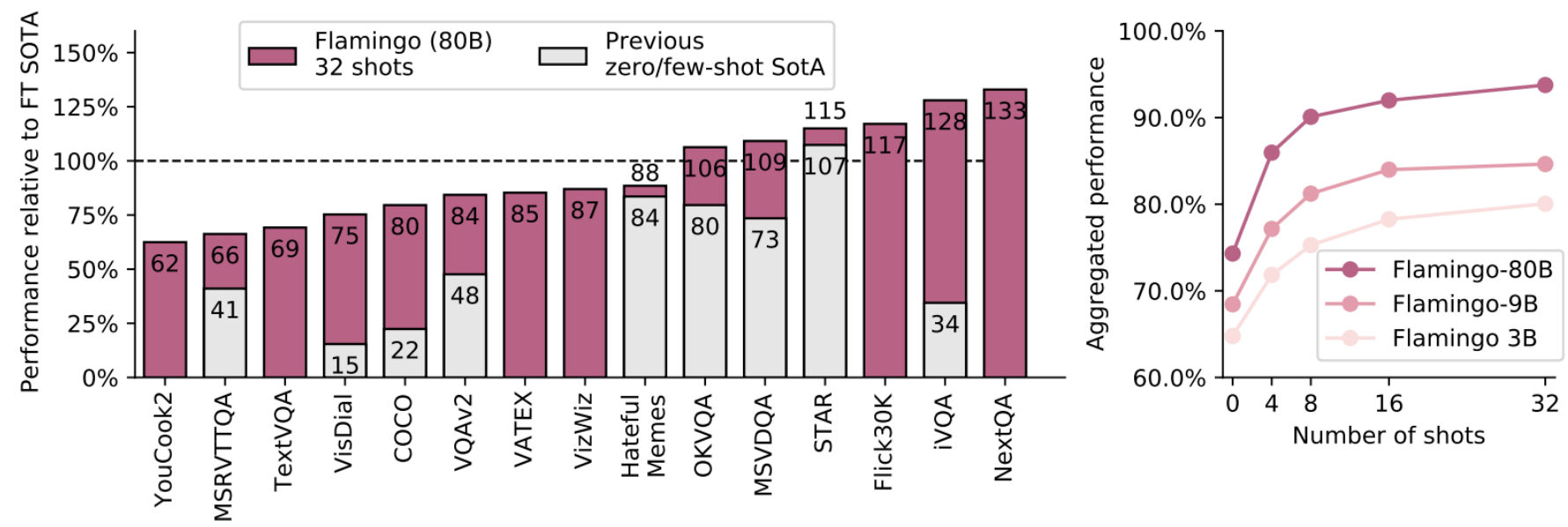
P I think it's Chicago because of the Shedd Aquarium in the background.



What about this one? Which city is this and what famous landmark helped you recognise the city?

P This is Tokyo. I think it's Tokyo because of the Tokyo Tower.

Flamingo results overview



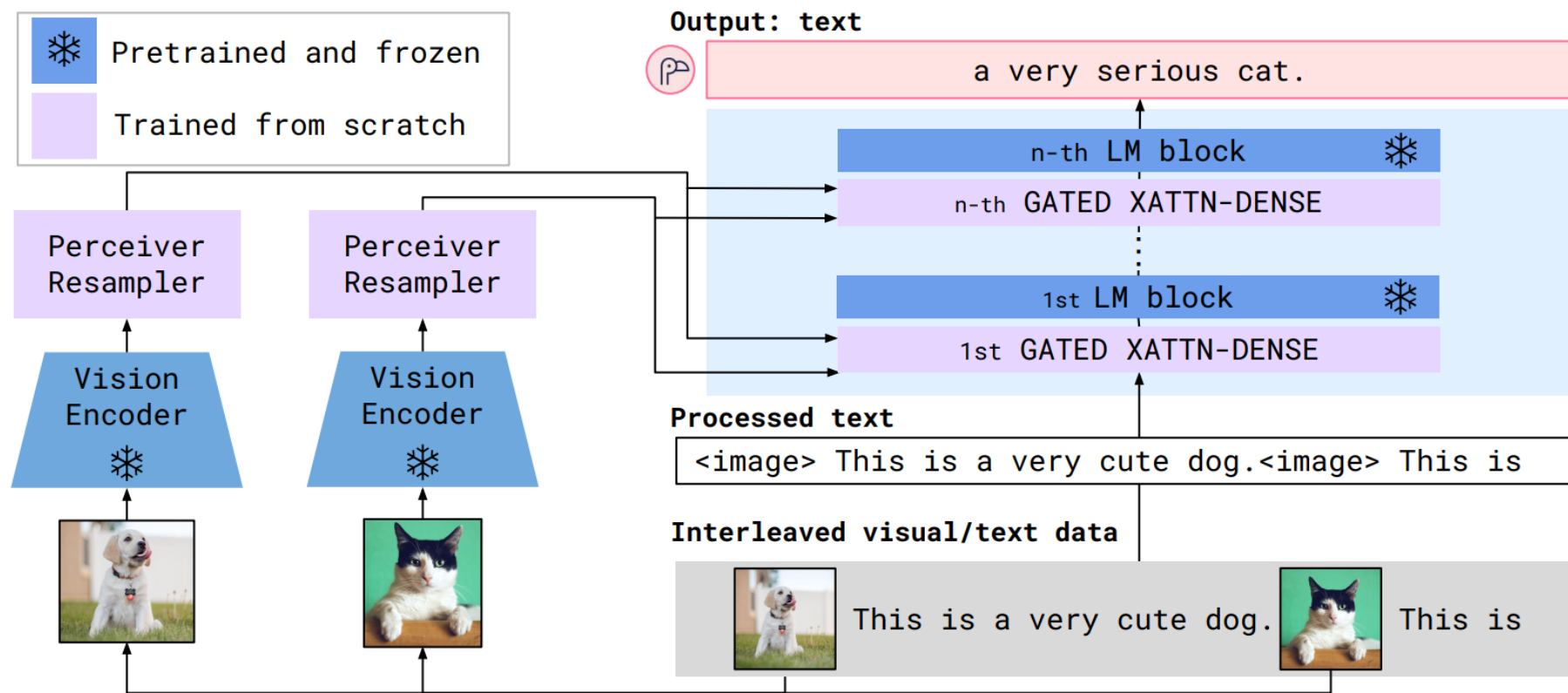
Introduction

- ✓ 현재 cv 모델은 대용량 데이터 셋으로 Pre-trained된 모델을 토대로 fine tuning하는 것이 추세이나 이는 수많은 양의 annotation data를 필요로 함
- ✓ 최근 vision-language model에서 zero-shot으로 모델을 적용하려는 시도가 있었지만 이는 classification과 같은 제한된 task에만 적용 가능
 - ✓ 이는 image captioning이나 VQA같은 text generation 부분에서 취약하다는 것을 뜻함

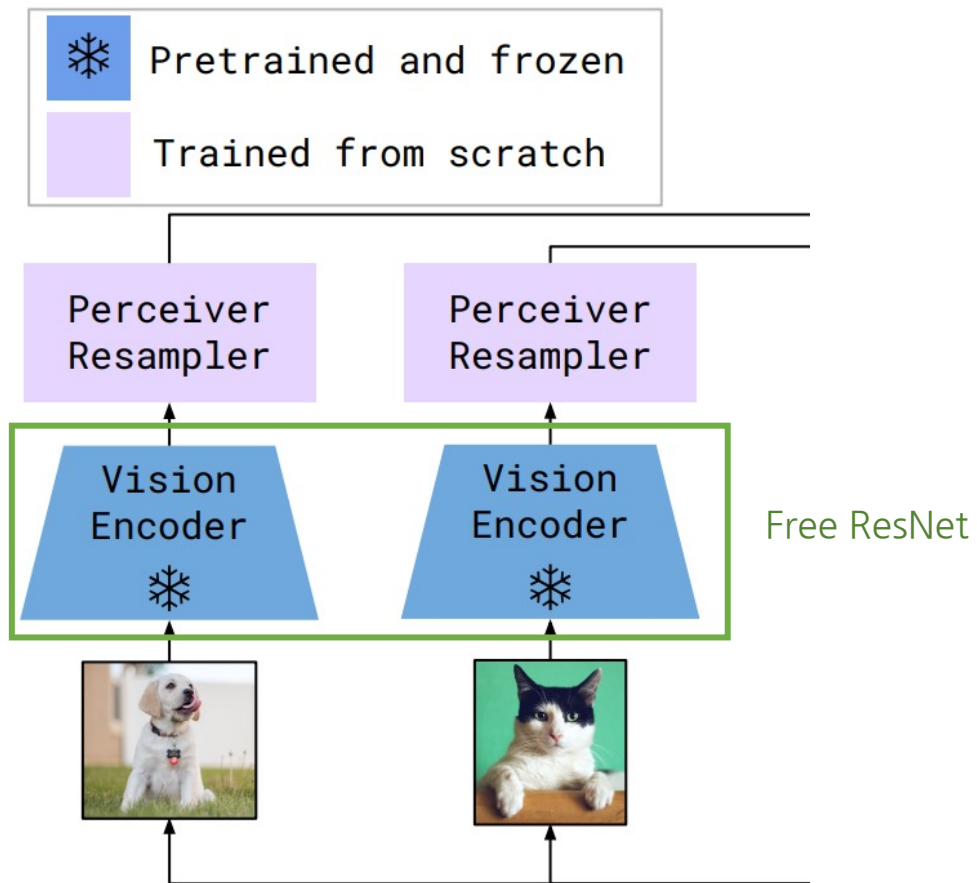
Introduction

- ✓ 현재 cv 모델은 대용량 데이터 세트로 Pre-training을 하는 것이 아니라, 적은 데이터 세트로 fine-tune을 하는 것이 목표
- ✓ 최근 vision-language model에서 zero-shot으로 모델을 적용하려는 시도가 있었지만 이는 classification과 같은 제한된 task에만 적용 가능
 - ✓ 이는 기존의 few-shot learning으로 좋은 성능을 냈던 LLM을 VLM에 적용하려는 시도
 - ✓ 이는 image captioning이나 VQA 같은 text generation 부분에서 취약하다는 것을 방증

Architecture

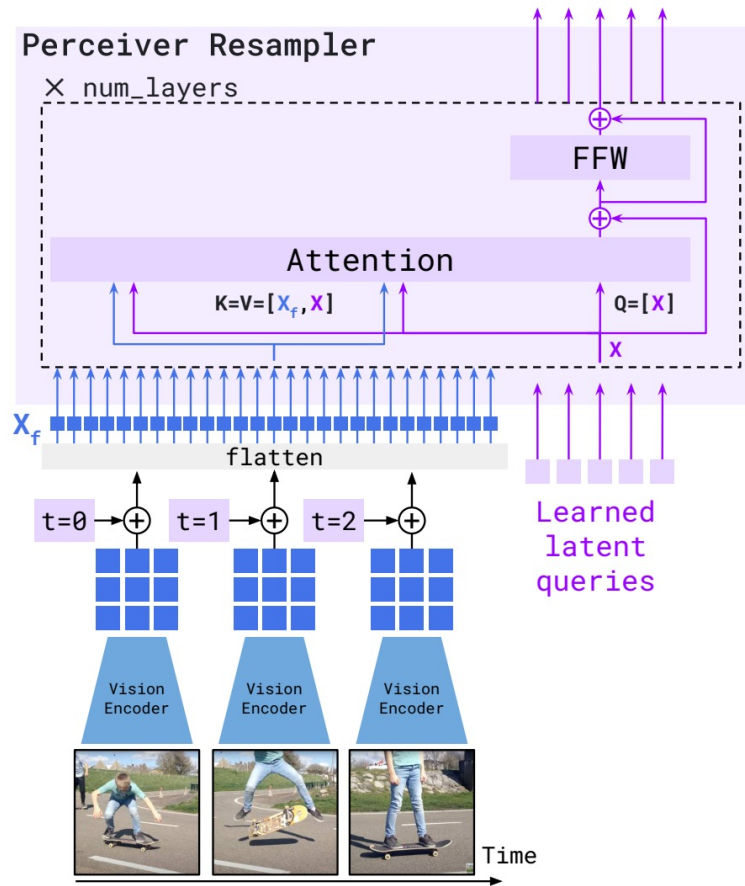


Architecture - vision Encoder



- ✓ Pre-trained된 Free ResNet 사용하여 Output은 flattened 1D - sequene
- ✓ 이 때 vision Encoder는 contrasive learning을 통해 color, shape, position 등의 semantic spatial feature 추출

Architecture - Perceiver Resampler

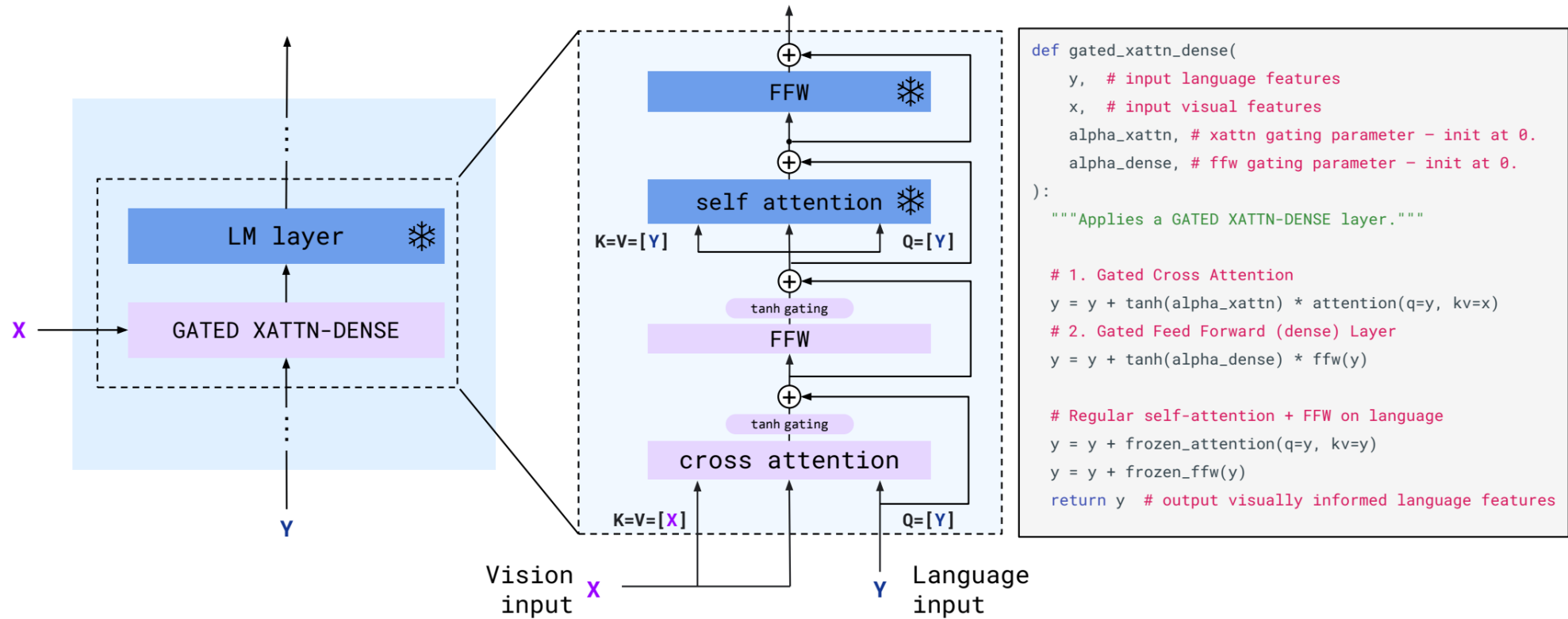


```
def perceiver_resampler(
    x_f, # The [T, S, d] visual features (T=time, S=space)
    time_embeddings, # The [T, 1, d] time pos embeddings.
    x, # R learned latents of shape [R, d]
    num_layers, # Number of layers
):
    """The Perceiver Resampler model."""

    # 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
```

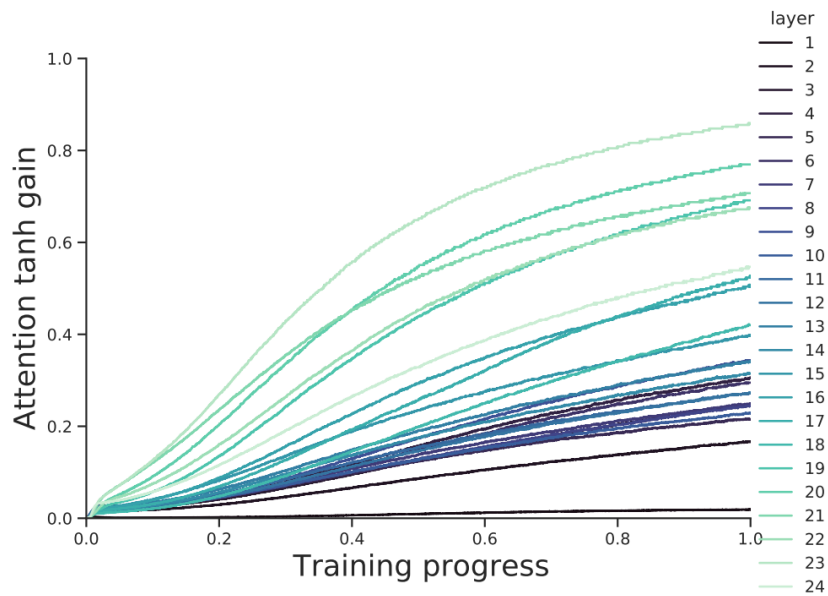
- ✓ 다양한 개수의 Vision Feature를 text Feature의 차원과 맞추기 위한 작업, 이 논문에서는 Learned latent queries와 같은 64로 맞춤
- ✓ flattened 1D - sequence가 key, value가 되며 learned latent queries와 cross attention

Architecture - Gated Cross Attention

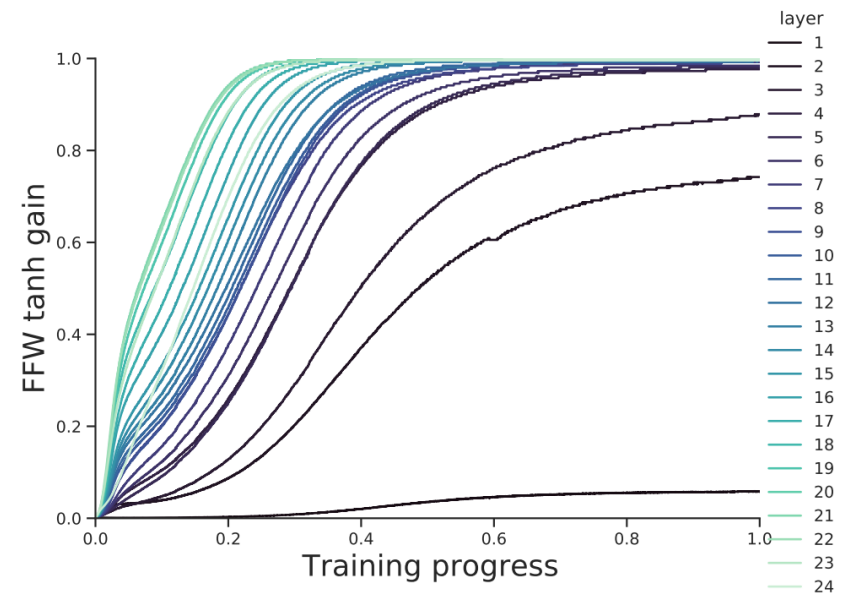


- ✓ Vision input을 key, value로 query를 language input으로 사용
- ✓ Residual connection과 tanh gating의 사용, 이때 값이 0부터 시작
- ✓ 학습이 진행됨에 따라 text model에서 vision language model로 변환

GATED XATTN-DENSE



(a) Attention tanh gating

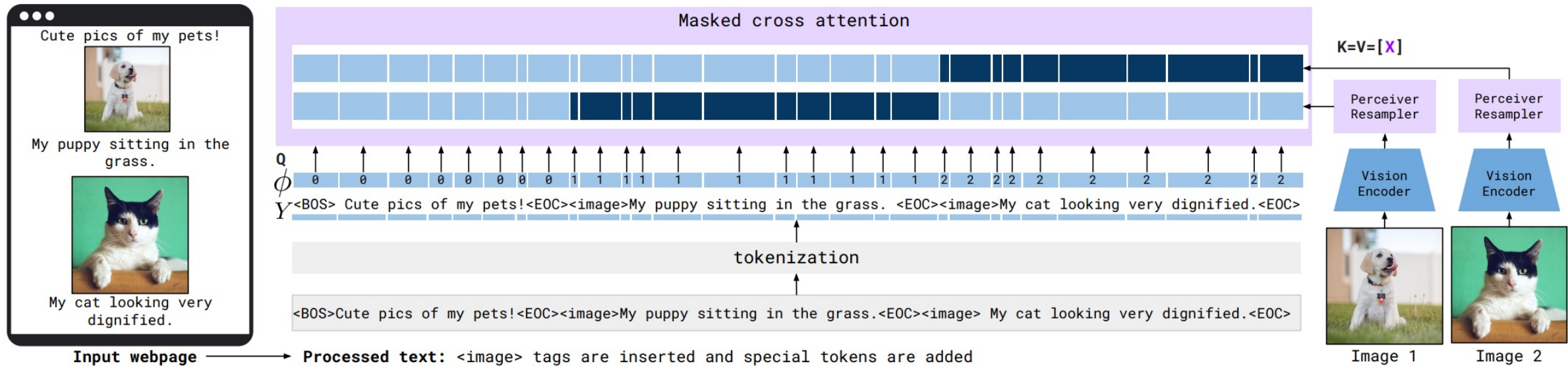


(b) FFW tanh gating.

✓ 후반 Layer로 갈수록 값이 높아지는 것을 확인

✓ 이는 처음에는 text에 대한 의미를 이해하는 것이 필수라는 것을 확인 가능

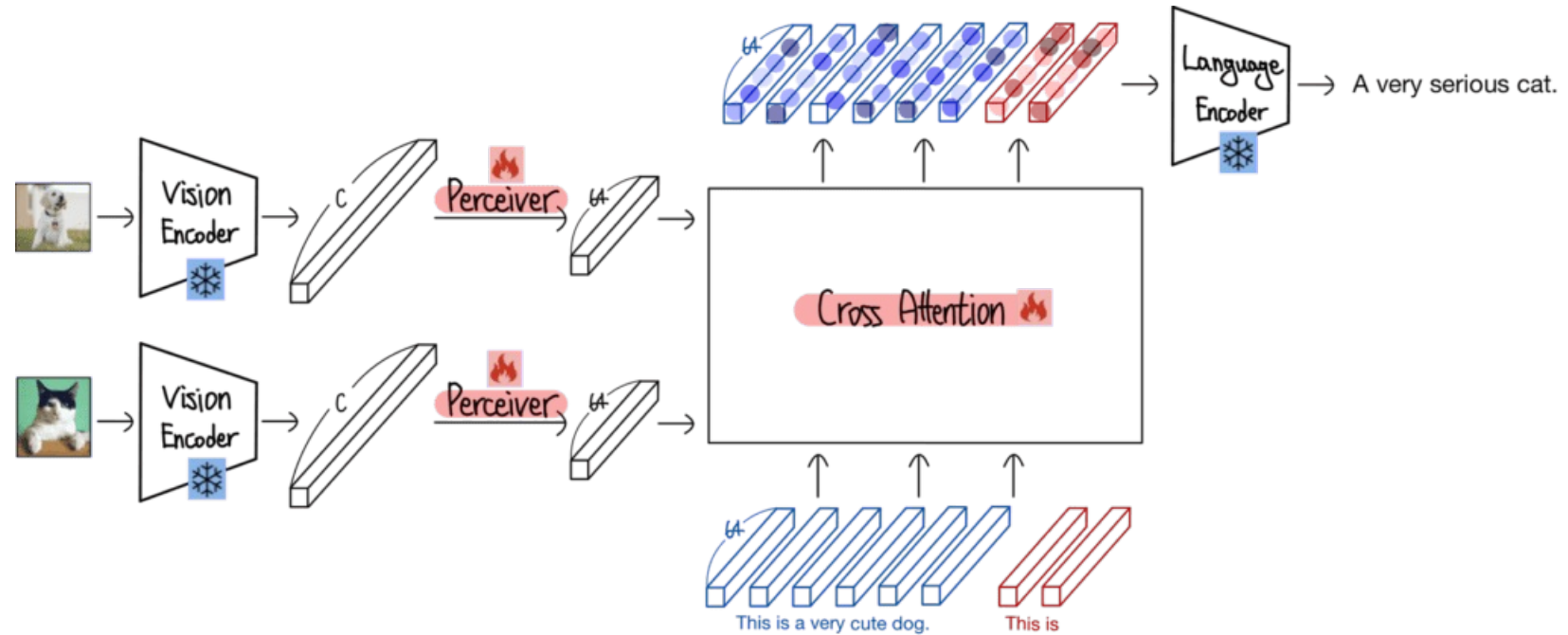
Interleaved visual data and text support



✓이미지에 맞는 text와 연산할 수 있게 인덱싱을 수행

✓이는 transformer에서 masking을 적용하는 것과 동일한 원리

Architecture



출처: <https://ffighting.net/deep-learning-paper-review/multimodal-model/flamingo/>

Training dataset details

- ✓ M3W collection: web상에서 크롤링 한 데이터로 약 43M개의 image, text 추출
 - ✓ ALIGN and LTIP: 약 1.8B개의 image와 text pair
 - ✓ VTP: 27M개의 비디오와 문장 설명의 pair로 구성

experiment

Method	FT	Shot	OKVQA (I)	VQA _{v2} (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	X	(X)	[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	X	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
	X	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	-
	X	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	X	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
	X	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	-
	X	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	X	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
	X	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	-
	X	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	✓	(X)	54.4	80.2	143.3	47.9	76.3	57.2	67.4	46.8	35.4	138.7	36.7	75.2	54.7	25.2	79.1	-
			[34] (10K)	[140] (444K)	[124] (500K)	[28] (27K)	[153] (500K)	[65] (20K)	[150] (30K)	[51] (130K)	[135] (6K)	[132] (10K)	[128] (46K)	[79] (123K)	[137] (20K)	[129] (38K)	[62] (9K)	

Ablation studies

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</i>-3B model				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

Conclusion

limitation

- ✓ Pre-trained LM에 의존하여 LM의 약점을 그대로 가져가게 됨
- ✓ Prompt에 민감하다
- ✓ Few-shot learning은 효과적이지만(<32) 그 이상의 데이터를 갖게 되면 연산량이 급격하게 상승할 수 있다

Conclusion

- ✓ 최소한의 task-specific data만으로 image/video tasks에 빠르게 적용 가능한 모델
- ✓ 기존 모델에 비해 상당한 flexibility 가짐

감사합니다 😊