Show, Attend and Tell:

Neural Image Caption Generation With Visual Attention (ICML2015)

”attention”

- Hard attention
- Soft attention
- Two attention based image caption generator model

1. image captioning model structure

overview

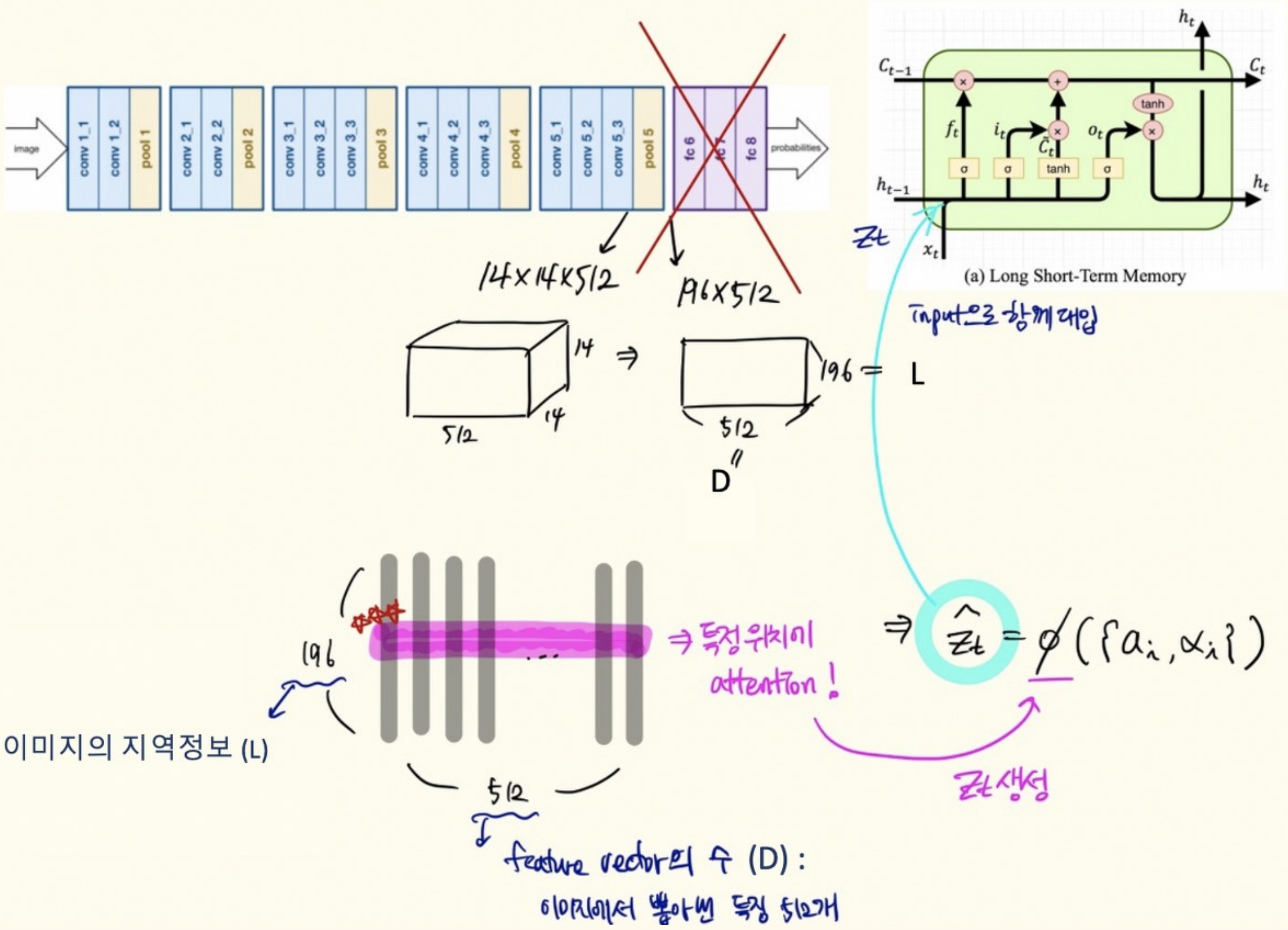


image
conv 1_1
conv 1_2
pool 1
conv 2_1
conv 2_2
pool 2
conv 3_1
conv 3_2
conv 3_3
pool 3
conv 4_1
conv 4_2
conv 4_3
pool 4
conv 5_1
conv 5_2
conv 5_3
pool 5
fc 6
fc 7
fc 8
probabilities
14x14x512
196x512
14
14
512
196 = L
512
|| D
(a) Long Short-Term Memory
tanh
Zt
Input으로 함께 대입
196
512
이미지의 지역정보 (L)
⇒ 특정 위치에 attention !
$\Rightarrow \hat{z}_t = \phi(\{a_i, \alpha_i\})$
Zt 생성
feature vector의 수 (D) :
이미지에서 뽑아낸 특징 512개

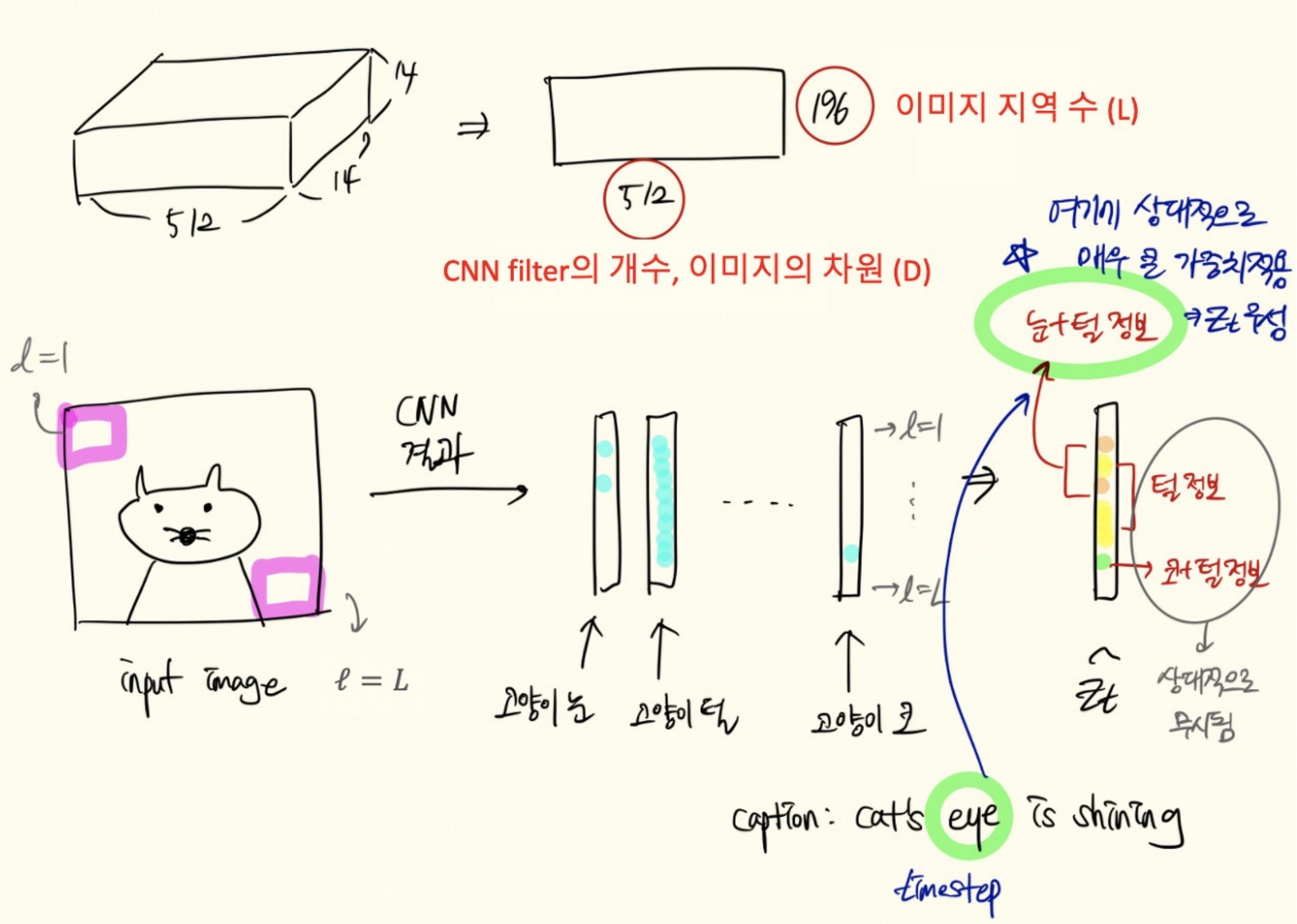
Encoder
(cnn)

Input: 주어진 이미지

Output: feature vector a (L*D)

- 마지막 layer은 l개의 필터와 d개의 뉴런
- Fc layer을 사용하지 않음

Encoder
(cnn)


14
14
512
196
이미지 지역 수 (L)
512
CNN filter의 개수, 이미지의 차원 (D)
여기에 상대적으로
매우 큰 가중치 적용
눈+털정보
→ ẑt 완성
l=1
CNN 결과
→l=1
→l=L
털정보
코+털정보
input image
l = L
고양이 눈
고양이 털
고양이 코
ẑt
상대적으로
무시됨
caption: cat's eye is shining
timestep

Decoder
(lstm)

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

기존 LSTM 수식

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix} \quad (1)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \quad (2)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \quad (3)$$

Attention 적용 후

Input:

1. Eyt-1 : 단어 embedding vector
2. ht-1 : hidden state vector
3. zt : image feature vector

2. Attention Mechanism

$$\operatorname{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boldsymbol{h}_t^\top \boldsymbol{W}_a \bar{\boldsymbol{h}}_s & \textit{general} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_a [\boldsymbol{h}_t ; \bar{\boldsymbol{h}}_s]\right) & \textit{concat} \end{cases}$$

1. Dot (내적): caption의 어느 부분이 이미지의 어느 위치 벡터와 가장 유사한지 파악
2. General: 두 벡터 사이의 연관도를 나타냄
3. Concat: 두 벡터를 concat한 후 학습

3. Stochastic Hard Attention

$$p(s_{t,i} = 1 \mid s_{j<t}, \mathbf{a}) = \alpha_{t,i}$$

→ timestep t에서 $s_{t,i}=1$일 확률 $\alpha_{t,i}$: multinoulli dist.

$$\hat{\mathbf{z}}_t = \sum_i s_{t,i}\mathbf{a}_i.$$

새로운 R.V (Random Variable) $\hat{z}_t$ 정의

$$s_t = [\underbrace{0, 0, 0, 1, \cdots, 0, \cdots, 0}_{\text{길이 } L}]$$

Why stochastic?

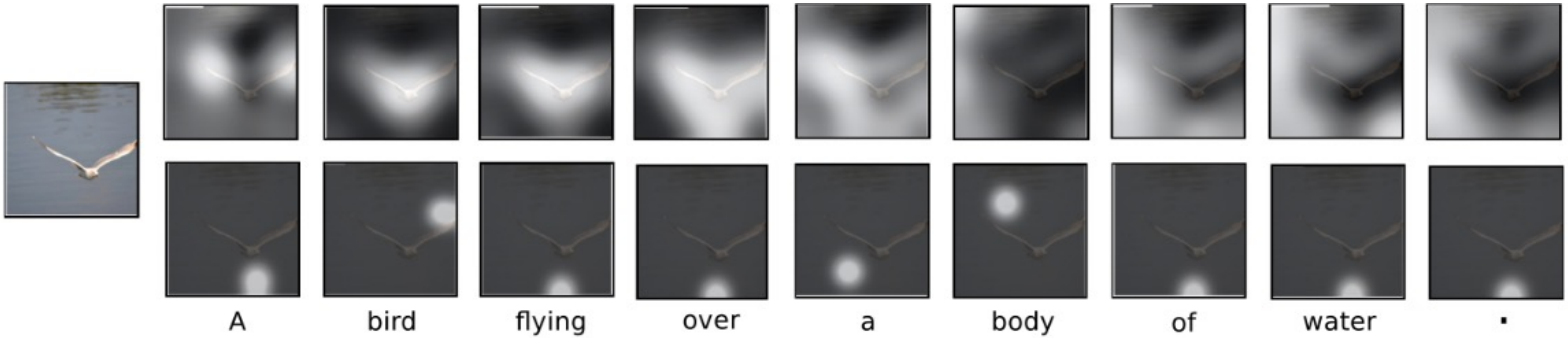
- 모든 s에 대해 계산을 하려면 즉, 모든 위치 i에 대해 모든 경우의 수를 다 계산하기는 어려움
- 따라서 mini batch와 비슷한 개념으로 sampling하여 gradient를 계산

4. Deterministic Soft Attention

$$\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t] = \sum_{i=1}^{L} \alpha_{t,i}\mathbf{a}_i$$

하나만 고르지 않고, 어느 것을 얼만큼 사용할 것인지 비율에 맞게 가져다 사용.
sampling하지 않아도 되고, end-to-end로 학습이 가능

4. 결론


A
bird
flying
over
a
body
of
water
.

Dataset	Model	BLEU				
		BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k	Google NIC(Vinyals et al., 2014)$^{\dagger\Sigma}$	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)$^{\circ}$	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k	Google NIC$^{\dagger\circ\Sigma}$	66.3	42.3	27.7	18.3	—
	Log Bilinear	60.0	38	25.4	17.1	16.88
	Soft-Attention	66.7	43.4	28.8	19.1	**18.49**
	Hard-Attention	**66.9**	**43.9**	**29.6**	**19.9**	18.46
COCO	CMU/MS Research (Chen & Zitnick, 2014)^{a}	—	—	—	—	20.41
	MS Research (Fang et al., 2014)$^{\dagger a}$	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014)$^{\circ}$	64.2	45.1	30.4	20.3	—
	Google NIC$^{\dagger\circ\Sigma}$	66.6	46.1	32.9	24.6	—
	Log Bilinear$^{\circ}$	70.8	48.9	34.4	24.3	20.03
	Soft-Attention	70.7	49.2	34.4	24.3	**23.90**
	Hard-Attention	**71.8**	**50.4**	**35.7**	**25.0**	23.04

- **BLEU score**: N-gram 단위로 정답 caption과의 유사도를 측정
- Flickr dataset 및 COCO dataset 모두에서 **attention을 적용한 모델의 성능(BLEU score)이 더 높음**