Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio

Delivered by 이현준, 2018.11.24

Executive Summary

이번 논문 **Show, Attend and Tell**에서는

1. **Visual Attention**이라는 개념을 Neural Image Captioning 문제에 적용하였습니다

2. Attention을 **Hard Attention**과 **Soft Attention**으로 구분하여 제안하였습니다

3. 기존 Neural Image Captioning이 가진 한계(Objectiveness)를 극복하였고,
다양한 Dataset에 대해 (당시 기준)SOTA의 성능을 달성하였습니다

1. Introduction

Caption Generation : Scene Understanding과 밀접한 주제

- Caption Generation의 두 가지 요소: **1) Object Detection 2) Expressing Relationship along object**
- Neural Image Caption Generation에서 각각 **1) Convolutional NN 2) Recurrent NN(LSTM)**으로 대응되어 왔음

Attention에 관하여

- Object Detection과 Feature Extraction을 위해 일반적으로 Convolutional NN을 주로 사용하였음
- 하지만 이미지 축약 과정에서 Captioning에 중요한 정보가 손실될 수 있고, 동적으로 중요 정보를 찾아낼 수 없음
- 정보손실을 방지하기 위해 Low-Level Representation을 사용하고, 필요에 의해(In need) 정보의 변경이 자유로워야 함
- **Attention Mechanism을 도입하면 이러한 문제를 해결할 수 있음**

1. Introduction

Main Contributions of this paper

1) 두 가지의 attention-based image caption generator를 제안하였음 – **'Hard' attention** & **'Soft' Attention**

- **Hard (Stochastic) Attention**: 근사적으로 variational lower bound를 극대화하거나 강화학습을 통해 구현이 가능함
- **Soft (Deterministic) Attention**: 대수적인 방법 (Backpropagation)을 통해 학습 및 구현이 가능함

2) Caption Generation 모델이 **참조하는 Feature를 시각화** 할 수 있음

- Annotation을 활용하여 Captioning을 위해 이미지의 어느 부분을 참조하는지 확인이 가능 (where and what)

3) 새로 제시된 모형의 성능을 계량화하여 평가하였고, SOTA를 달성함

- 3개의 Dataset (Flickr8k, Flickr30k, MS COCO)에 대해 SOTA를 달성

2. Related Work

RNN(LSTM)을 활용한 Seq2Seq 기계번역 방법론에서 많은 영향을 받음

- 주로 **Learning Phrase Representation using RNN Encoder-Decoder for Statistical Machine Translation**(풀잎 3주차)을 참고한 것으로 보임
- 기존 연구 (**Show and Tell** 포함)도 LSTM을 차용해 Image captioning 문제를 해결해왔음
- **Image로부터 추출한 Feature를 Sentence로 번역하는 문제**로, 기계번역과 같은 방식으로 해결하려는 아이디어가 핵심

Object Detection에 대하여

- 기존 연구는 주로 object detector를 학습시킨 후, 그 결과를 학습된 language model에 적용하는 방식을 사용
- 이 연구에서는 object detector를 따로 사용하지 않고, **Latent Alignment**를 학습시키는 방식을 사용
- Latent Alignment를 사용하면 목표성(Objectiveness)을 넘어 포괄적이고 고차원적인 개념을 포착(attend)할 수 있음

3. Image Caption Generation with Attention Mechanism

3.1 Model Details

3.1.1 Encoder: Convolutional Features

- $\mathbf{y}$: Model의 최종 결과물인 Sequence of encoded words

$$\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_C\}, \qquad \mathbf{y}_i: \text{인코딩된 단어}, \in \mathbb{R}^K, \qquad C: \text{캡션의 길이}, \qquad K: \text{Vocabulary size}$$

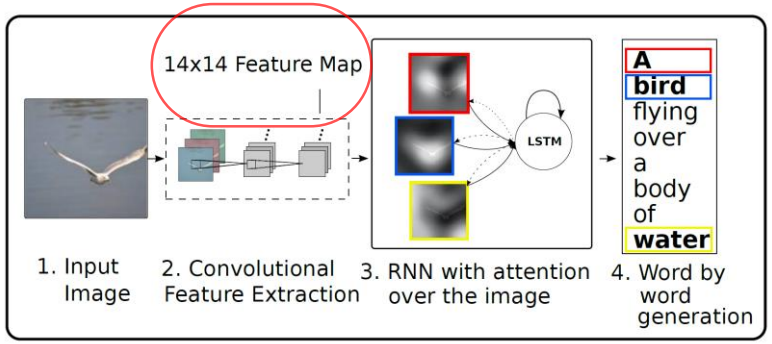
- $\mathbf{a}$: Encoder를 통과한 Image의 feature, **annotation 정보를 포함하고 있음** (그래서 $\mathbf{a}$)

$$\mathbf{a} = \{\mathbf{a}_1, \dots, \mathbf{a}_L\}, \qquad \mathbf{a}_i \in \mathbb{R}^D,$$

L: extractor가 뽑아낸 vector 숫자(annotation, 즉 attention이 참조할 좌표값)

D: 각 vector의 representation dimension (즉 CNN의 Filter Number)

- 기존 모델과의 차별점: low-level convolutional layer를 사용함 (논문에서 구현한 최종 feature는 **14X14**)



3. Image Caption Generation with Attention Mechanism

3.1 Model Details

3.1.2 Decoder: Long Short-Term Memory Network (LSTM)

- 논문에서는 LSTM cell의 연산을 Vector form으로 통합해서 표현함 (Zaremba et al., 2014)
- 여타 LSTM과 동일하게 연산에 필요한 요소를 Inputs / Gates / Outputs로 구분할 수 있음

1) Inputs: $\mathbf{E}_{\mathbf{y}_{t-1}}$, $\mathbf{h}_{t-1}$, $\hat{\mathbf{z}}_t$ (3개)

· $\mathbf{E}_{\mathbf{y}_{t-1}}$: 직전 timestep에 출력된 단어 $\mathbf{y}_{t-1}$의 Embedding Matrix (m by K) – 기존에 출력된 단어 정보들을 담고 있음

· $\mathbf{h}_{t-1}$: 직전 timestep LSTM cell 연산 결과물인 hidden state – LSTM 연산을 통해 축적된 정보를 담고 있음

· $\hat{\mathbf{z}}_t$: 현 timestep에서 계산된 context vector – Feature $\mathbf{a}$에서 참조할 attention의 정보를 담고 있음

$$\hat{\mathbf{z}}_t = \phi(\{\mathbf{a}_i\}, \{\alpha_i\})$$

· $\boldsymbol{\phi}$: single vector를 출력하는 연산, input으로 $\mathbf{a}_i$와 α_i의 쌍을 받음

· $\boldsymbol{\alpha_{ti}} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L}\exp(e_{tk})}$ where $e_{ti} = attention(\mathbf{a}_i, \mathbf{h}_{t-1})$

즉 α_{ti}는 t번째 단어를 출력할 때 필요한 weight이며, Convolutional Feature의 좌표 i에 부여되는 attention의 비중이다

· 그러므로 $\hat{\mathbf{z}}_t$는 Image Feature의 **'어느 부분을 얼마만큼' Caption Generation에 사용할지 결정하는 Context**를 담고있다!

Reference: **SanghyukChun's Blog**, http://sanghyukchun.github.io/93/

3. Image Caption Generation with Attention Mechanism

3.1 Model Details

3.1.2 Decoder: Long Short-Term Memory Network (LSTM)

2) Gates: i_t, g_t, f_t, o_t (4개)

· i_t **:** input gate, cell state $\mathbf{c}_t$에 저장할 input value들의 비중을 결정함 (sigmoid)

· g_t **:** input modulator, cell state $\mathbf{c}_t$에 저장할 input value들의 값을 결정함 (tanh)

· f_t **:** forget gate, cell state $\mathbf{c}_t$에서 제외할 이전 cell state $\mathbf{c}_{t-1}$의 비중을 결정함 (sigmoid)

· o_t **:** output gate, 출력할 hidden state $\mathbf{h}_t$ 에 포함할 cell state $\mathbf{c}_t$의 비중을 결정함 (sigmoid)

3) Outputs: $\mathbf{c}_t, \mathbf{h}_t$ (2개)

· $\mathbf{c}_t = \mathrm{f}_t \odot \mathbf{c}_{t-1} + \mathrm{i}_t \odot \mathrm{g}_t$, 이전 cell state의 정보를 지우고/지우지 않고, 새로운 input의 정보를 받아들임/들이지 않음

· $\mathbf{h}_t = \mathrm{o}_t \odot \tanh(\mathbf{c}_t)$, 현재 cell state의 정보를 반영하여 hidden state를 출력함

4) 그 외

· $T_{D+m+n,n}$: $\mathbb{R}^{D+m+n} \rightarrow \mathbb{R}^n$의 simple affine transformation, 즉 $\mathbb{R}^n$으로 변형시켜주는 weight matrix를 곱하는 연산임

Reference: **SanghyukChun's Blog**, http://sanghyukchun.github.io/93/

3. Image Caption Generation with Attention Mechanism

3.1 Model Details

3.1.2 Decoder: Long Short-Term Memory Network (LSTM)

5) LSTM cell 도식화

gates

n-dimension 으로 변환

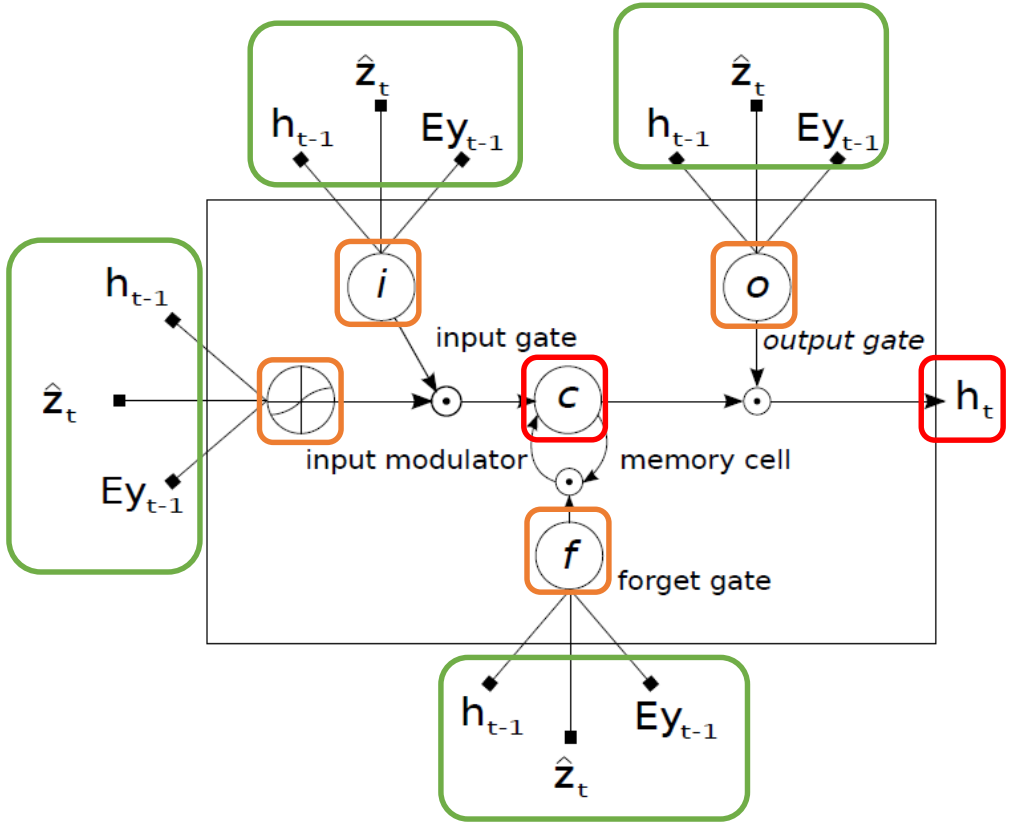
$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix}$$

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

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

inputs

outputs



Reference: **SanghyukChun's Blog**, http://sanghyukchun.github.io/93/

3. Image Caption Generation with Attention Mechanism

3.1 Model Details

3.1.2 Decoder: Long Short-Term Memory Network (LSTM)

6) output 초기화 $(\mathbf{c}_0, \mathbf{h}_0)$

· 최초로 LSTM cell을 구현하기 위해 필요한 $\mathbf{c}_t, \mathbf{h}_t$의 초기값을 설정해야 함

· 논문에서는 annotation vector(좌표값)의 평균에 대해 초기화 연산을 진행함 (MLP)

7) inputs를 활용한 가능도함수(목적함수) 표현

· 일반적으로 LSTM을 활용한 language model은 이전에 출력된 단어들과 input 정보를 반영하여 다음 단어가 생성될 가능도함수를 최대화(Maximum Likelihood Principle)하는 방식으로 모형을 구현함

· 이 논문에서는 Deep output layer라는 다층 가중치행렬을 사용하여 가능도함수를 아래와 같이 근사함

· 이 때 LSTM의 input들을 사용하여 가능도함수를 표현하는 장점이 있으며, **각 input에 부여되는 가중치행렬을 학습**하여 Loss Function을 최소화, 또는 목적함수를 최대화할 수 있게 됨!

$$p(\boldsymbol{y}_t | \boldsymbol{a}, \boldsymbol{y}_1^{t-1}) \propto \exp\left(\mathbf{L}_o\left(\mathbf{E}_{\mathbf{y}_{t-1}} + \mathbf{L}_h \mathbf{h}_{t-1} + \mathbf{L}_z \hat{\mathbf{z}}_t\right)\right)$$

$\mathbf{L}_o$, $\mathbf{L}_h$, $\mathbf{L}_z$, $\mathbf{E}$: **학습에 사용되는 파라미터들!** (최종 weight인 $\mathbf{L}_o$를 사용해 단어공간 $\mathbb{R}^K$로 mapping함)

Reference: **SanghyukChun's Blog**, http://sanghyukchun.github.io/93/

4. Learning Stochastic “Hard” vs Deterministic “Soft” Attention

※ 참고

1) 결정론적 모형 (Deterministic Model)

· 결정론적 모형은, 모형의 결과값이 **모수(Parameter)**와 **최초로 주어진 조건(initial conditions)**에 의해 온전히 결정되는 모형을 말함

· 즉 단순한 함수 ($y = ax + b$)나 FNN ($\mathbf{WX} + \mathbf{b}$)등 변수와, 변수를 표현하기 위한 parameter로 이루어진 관계들은 결정론적 모형의 일종

· 그러므로 결정론적 모형에 대한 모수를 찾기 위해서 **SGD와 같은 대수연산(algebra)을 사용**할 수 있음

2) 확률론적 (비결정론적) 모형 (Stochastic Model)

· 확률론적 모형은 변수 혹은 모수(Parameter)들이 임의성(Randomness)을 포함하고 있음

· 그 결과로 모수와 초기 조건들은 특정 값들의 집합과, 각 집합 요소에 mapping된 가능도(확률)에 의해 결정됨

· 확률론적 모형을 구성하는 모수를 추정하기 위해서는 최대 가능도 추정(Maximum Likelihood Estimation; **MLE**), 최대화 사후확률(Maximum a posteriori; **MAP**), 변분 베이지안 방법 (**Variational Bayesian Method**)등이 사용됨

4. Learning Stochastic “Hard” vs Deterministic “Soft” Attention

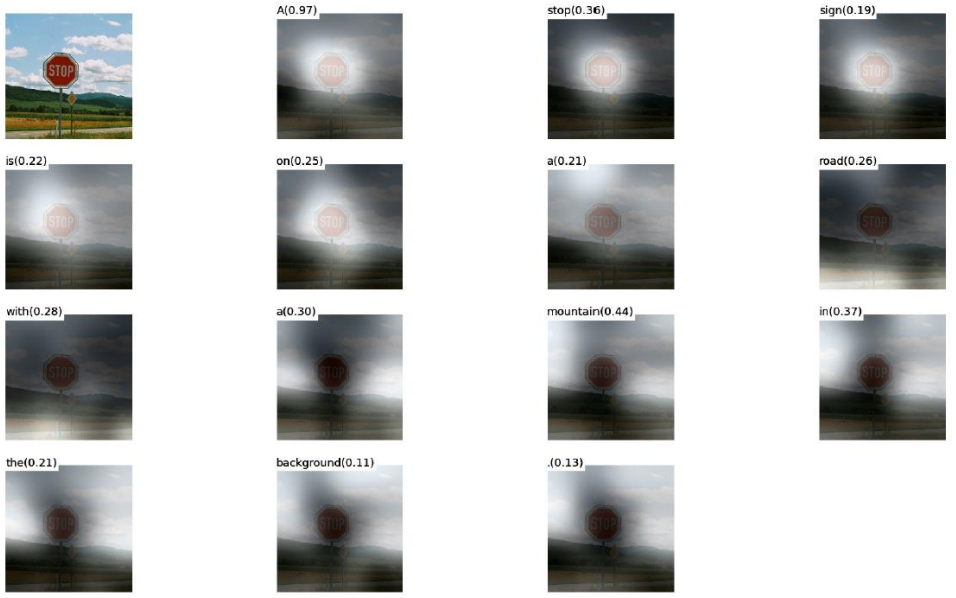
※ 두 모형의 Attention 예시

1) Stochastic “Hard” Attention

(a) A stop sign with a stop sign on it.

- Attention이 가상 구획의 ‘한 지점’에만 발생
- context vector를 결정하는 변수는 one-hot 변수

2) Deterministic “Soft” Attention



(b) A stop sign is on a road with a mountain in the background.

- Attention이 구획별 ‘가중치’의 형태(음영)로 발생
- context vector를 결정하는 변수는 확률(softmax)

4. Learning Stochastic “Hard” vs Deterministic “Soft” Attention

4.1 Stochastic “Hard” Attention

1) 정의

- $s_{t,i}$: t번째 단어를 생성할 때 feature annotation i를 참조했는지 여부를 판별하는 변수 (0 or 1)

- $\alpha_{t,i} = p(s_{t,i} = 1 | s_{j<t}, \mathbf{a})$: $s_{t,i}$가 1인 경우 (i번째 좌표를 attend한 경우)의 확률값을 부여하는 ‘확률모수’(parameter)

- $\hat{\boldsymbol{z}}_t = \sum_i s_{t,i} * \mathbf{a}_i$, annotation i와, i를 참조했는지 여부를 판별하는 ‘확률변수’ 의 가중치 합계

  → Image Feature의 **‘어느 부분을 얼마만큼’을 결정하는 Context**라는 앞선 정리와 일치!

2) Stochastic (Hard) Attention-based Model의 목적함수 도출

- 모형의 목적은 로그 가능도함수 $\log(p(\mathbf{y}|\mathbf{a}))$를 최대화하는 것임 (feature가 주어졌을 때의 caption의 등장 가능성)

- 한편 변분 베이지안 방법을 활용하면, 로그 가능도함수의 변분 하한값(Variational Lower Bound)를 찾는 것이 가능함
  그러므로 가능도함수 최대화 문제는 **변분 하한값 최대화 문제로 치환하여 해결**하는 것이 가능함

- $\log p(\mathbf{y}|\mathbf{a}) \geq \sum_s p(s|\mathbf{a}) \log p(\mathbf{y}|\mathbf{s}, \mathbf{a}) := \boldsymbol{L_s}$(변분 하한값) (*by variational Bayesian method*)

- 이 $\boldsymbol{L_s}$이 **Hard attention-based model의 최대화 목적함수(또는 Loss Function)가 됨**

4. Learning Stochastic “Hard” vs Deterministic “Soft” Attention

4.1 Stochastic “Hard” Attention

3) 목적함수 최적화 문제

- L_s에서 변수를 제외한 모든 모수(parameter)를 W로 표현하면 최적화 문제는 아래의 식을 0으로 만드는 문제

$$\frac{\partial L_s}{\partial W} = \sum_s p(s|\mathbf{a}) * \left[\frac{\partial \log p(\mathbf{y}|s,\mathbf{a})}{\partial W} + \log p(\mathbf{y}|s,\mathbf{a}) * \frac{\partial \log p(s|\mathbf{a})}{\partial W}\right]$$

- Monte Carlo based Sampling을 통해, 변수 s를 확률변수 $\tilde{s_t} \sim Multinoulli(\{\alpha_i\})$로 근사할 수 있음 (결과는 논문의 식 (12))
- 목적함수를 최대화하는 변분 추정량의 분산을 줄이고, 모형의 강건성(Robustness)을 위해 아래와 같은 방법을 추가
  - · Moving average $\boldsymbol{b_k} = 0.9 * b_{k-1} + 0.1 * \log p(\mathbf{y}|\tilde{s_k}, \mathbf{a})$, 이전 로그 가능도의 exponential decay를 사용한 합으로 표현함
  - · Entropy term $\boldsymbol{H}[s]$을 추가하고, 0.5의 확률로 $\tilde{s}$의 값을 기대값인 α로 설정
- 이 방법을 모두 적용한 논문의 최종 근사 변분추정량(Parameter 값) 계산식은 아래와 같이 제시됨

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N}\sum_{n=1}^{N}\left[\frac{\partial \log p(\mathbf{y}|\tilde{s}^{\boldsymbol{n}},\mathbf{a})}{\partial W} + \lambda_r * [\log p(\mathbf{y}|\tilde{s}^{\boldsymbol{n}},\mathbf{a}) - b] * \frac{\partial \log p(\tilde{s}^{\boldsymbol{n}}|\mathbf{a})}{\partial W} + \lambda_e * \frac{\partial H[\tilde{s}^{\boldsymbol{n}}]}{\partial W}\right]$$

- 위 식이 강화학습의 update rule과 같기 때문에, hard attention model은 강화학습으로도 학습이 가능함
- Attention의 목적인 $\hat{\boldsymbol{z}}_t = \sum_i s_{t,i} * \mathbf{a}_i$는 $\tilde{s_t} \sim Multinoulli(\{\alpha_i\})$로부터 $\boldsymbol{s_{t,i}}$**를 추정(Sampling)하여 계산**하게 됨

4. Learning Stochastic “Hard” vs Deterministic “Soft” Attention

4.2 Deterministic “Soft” Attention

1) “Hard” Attention과 다른점

- “Hard” Attention에서는 $\hat{\boldsymbol{z}}_t(=\sum_i s_{t,i} * \mathbf{a}_i)$를 계산하기 위해 $s_{t,i}$를 모수모형으로 만들어 추정하는 방식을 사용
- 반면 “Soft” Attention은 $\hat{\boldsymbol{z}}_t$의 기대값인 $\mathbb{E}_{p(s_t|a)}[\hat{\boldsymbol{z}}_t](=\sum_i \alpha_{t,i} * \mathbf{a}_i)$를 추정에 사용

  → 앞서 정의된 $\boldsymbol{\phi}(\{\mathbf{a}_t\},\{\alpha_t\})$를 $\sum_i \alpha_{t,i} * \mathbf{a}_i$로 설정하면 각 $\boldsymbol{\alpha_t}$를 **모수로 하는 결정론적 모형의 최적화 문제**로 해결 가능

  → $\boldsymbol{\alpha_t}$에 대한 **SGD (Backpropagation)을 사용할 수 있게 됨!**

2) Deterministic (Soft) Attention-based Model의 목적함수 도출

- Soft Attention Model의 목적함수를 도출하기 위해 몇 가지의 전제조건이 필요함
  - $\mathbb{E}_{p(s_t|a)}[\mathbf{h}_t]$ (hidden state의 기대값) : $\mathbf{h}_t$ 는 $\hat{\boldsymbol{z}}_t$의 linear projection, 그러므로 $\mathbb{E}_{p(s_t|a)}[\mathbf{h}_t]$의 1차 Taylor 근사값은 사영의 기대값인 $\mathbb{E}_{p(s_t|a)}[\hat{\boldsymbol{z}}_t]$을 활용한 연산과 같아짐
  - $\mathbf{n}_t = \mathbf{L}_o\left(\mathbf{E}_{\mathbf{y}_{t-1}} + \mathbf{L}_h\mathbf{h}_{t-1} + \mathbf{L}_z\hat{\mathbf{z}}_t\right)$ (가능도함수의 근사식), $\mathbf{n}_{t,i}$는 i번째 annotation $\mathbf{a}_i$에 대한 계산 결과라 한다

4. Learning Stochastic "Hard" vs Deterministic "Soft" Attention

4.2 Deterministic "Soft" Attention

2) Deterministic (Soft) Attention-based Model의 목적함수 도출 (계속)

- 전제조건하에서, 가능도함수 $p(\mathbf{y}_t = k|\mathbf{a})$ 정규화 가중 기하평균(NWGM)은 다음과 같이 정의함

$$NWGM[p(\mathbf{y}_t = k|\mathbf{a})] = \frac{\exp\left(\mathbb{E}_{p(s_t|a)}[\mathbf{n}_{t,k}]\right)}{\sum_j \exp\left(\mathbb{E}_{p(s_t|a)}[\mathbf{n}_{t,j}]\right)}, \qquad \mathbb{E}_{p(s_t|a)}[\mathbf{n}_t] = \mathbf{L}_o\left(\mathbf{E}_{\mathbf{y}_{t-1}} + \mathbf{L}_h\mathbb{E}_{p(s_t|a)}[\mathbf{h}_{t-1}] + \mathbf{L}_z\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t]\right)$$

- 정의된 NWGM은 softmax activation 하에서 $\mathbb{E}[p(\mathbf{y}_t = k|\mathbf{a})]$로 근사할 수 있음
- 또한 $\mathbb{E}_{p(s_t|a)}[\mathbf{h}_{t-1}]$ 은 전제조건에 의해 $\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t]$으로 표현될 수 있음

  → 그러므로 가능도함수 $p(\mathbf{y}_t = k|\mathbf{a})$의 기대값은 $\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t]$에 관한 식으로 근사할 수 있게 되어
  attention location에 관한 정보로 완전히 표현할 수 있게 됨 (**간단한 Feedforward propagation 활용 가능!**)

- 추가로 논문에서는 Doubly Stochastic Attention이란 개념을 같이 제안함 (자세한 설명 생략)
- **모든 사항을 고려한 Soft Attention-based Model의 최소화 대상 목적함수는 아래와 같음**

$$\boldsymbol{L_d} = -\log\left(p(y|x)\right) + \lambda\sum_i^L\left(1 - \sum_t^C \alpha_{ti}\right)^2$$

4. Learning Stochastic “Hard” vs Deterministic “Soft” Attention

4.3 Training Procedure

논문에서는 두 Attention-based Model 학습에 모두 SGD를 사용함

- 최적화 방법은 Datasets마다 다르게 적용 (Flickr8k – RMS prop / Flickr30k, MS COCO – ADAM)

이미지에서 Feature(annotation)을 추출하기 위해 pre-trained VGGnet을 사용함

- Feature map으로는 14*14*512를 사용 (**즉 L=14*14, D=512**)

Dropout과 early stopping만을 사용하였음

5. Experiments

5.1 Data & 5.2 Evaluation Procedure

논문에서는 총 3개의 Dataset을 사용하였음

- Flickr8k: 8,000장의 이미지와 이미지당 5개의 예시 문장이 mapping되어있음
- Flickr30k: 30,000장의 이미지와 이미지당 5개의 예시 문장이 mapping되어있음
- MS COCO: 82,783장의 이미지와 이미지당 5개, *또는 그 이상*의 예시 문장이 mapping되어있음
- 모든 dataset에 대해 vocabulary size V는 10,000으로 고정하였음
- 평가 척도로는 BLUE 1,2,3,4와 METEOR 척도의 5개를 사용하였음

정확한 평가를 위해 아래와 같은 조건을 고려하였음

- 기존 연구와의 형평성을 위해 Encoder network로 VGG / GoogLeNet을 사용하였음
  (METEOR 척도만 예외적으로 AlexNet 사용 결과를 차용함)
- **Model Ensemble을 사용하지 않고** 단독 모형을 사용한 결과만을 비교함
- Training / Test set의 standard split이 없는 경우 선행 연구를 참고하여 분할하였음

5. Experiments

5.3 Quantitative Analysis

두 개의 Attention-based Model 모두 뛰어난 성능을 보임

- MS COCO의 경우, 앞서 기술한 여러 technique이 METEOR 점수를 크게 향상시켰음
- Ensemble을 사용하지 않고도 좋은 결과를 보인 것이 특기할 만 함

Table 1. BLEU-1,2,3,4/METEOR metrics compared to other methods, † indicates a different split, (—) indicates an unknown metric, ○ indicates the authors kindly provided missing metrics by personal communication, Σ indicates an ensemble, a indicates using AlexNet

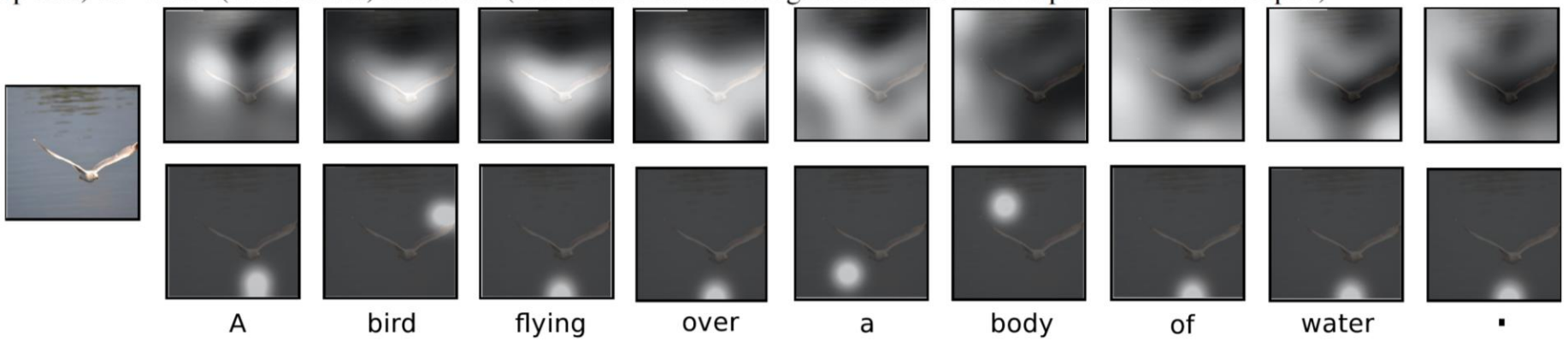
		BLEU				
Dataset	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k	Google NIC(Vinyals et al., 2014)†Σ	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)○	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k	Google NIC†○Σ	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)†^{a}	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014)○	64.2	45.1	30.4	20.3	—
	Google NIC†○Σ	66.6	46.1	32.9	24.6	—
	Log Bilinear○	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

5. Experiments

5.4 Qualitative Analysis: Learning to attend

기존 모델과 달리 모형이 참조하는 이미지를 Visualize할 수 있음

Figure 2. Attention over time. As the model generates each word, its attention changes to reflect the relevant parts of the image. "soft" (top row) vs "hard" (bottom row) attention. (Note that both models generated the same captions in this example.)



Objectiveness로부터 벗어나 'Non object' (배경 등) region을 attend하고 표현할 수 있음

End of Document

Thank you for your attention