😀

(2014) Show and Tell: A Neural Image Caption Generator

Abstract

해당 논문에서는 이전까지 자동으로 이미지를 설명하는 모델의 어려움을 설명하였습니다. 그리고 해당 논문에서는 컴퓨터 비전과 기계 번역을 통합한 하나의 일반화된 모델을 제안하고 있습니다. 해당 모델의 경우 이미지에 대한 설명을 liklihood를 최대화 하도록 학습하였습니다. 이를 통해서 BELU 점수에서 이전의 모델들에 비해 높은 성능을 보여주었습니다.

정리 : 이전에는 이미지를 자동으로 설명하는 모델의 어려움이 존재했다, 그래서 비전과 기게번역을 결합한 하나의 통합 모델을 제안한다.

Introduction

기존의 이미지에서 분류, 객체 탐지의 경우 단순히 이미지 특징 추출만 가능하지만 이미지를 설명하는 Task는 더욱 어렵습니다. 그리고 이전의 연구들은 이들을 하위 Task(단어 재배열, 개별 단어 번역)로 분리하여 여러 파이프라인을 통해서 학습할 수 있도록 하였습니다. 하지만 해당 논문에서는 이미지 I가 주어진 경우 $p(S|I)$ 의 우도를 최대화 함으로써 최종적으로 $S = [s_1, s_2, ...s_n]$ 을 얻는 모델을 제안합니다. 기존의 기계번역이 입력값과 타겟값의 $P(T|S)$을 최대화 하는 방식에서 영감을 받았습니다. 추가적으로 최근에는 RNN의 'encoder'는 문장의 의미를 벡터로 표현해주고, 'decoder'가 벡터를 통해 문장을 생성하는 기술이 발달 하여 보다 효과적일 것으로 기대하고 있습니다.

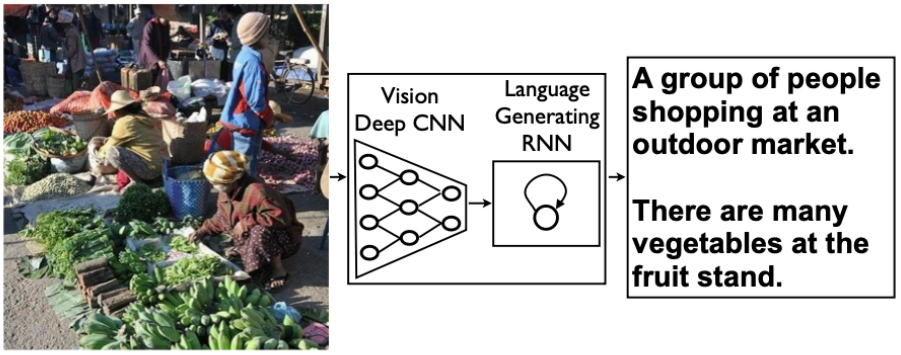


Figure 1. NIC, our model, is based end-to-end on a neural network consisting of a vision CNN followed by a language generating RNN. It generates complete sentences in natural language from an input image, as shown on the example above.

해당 논문에서 기존의 RNN의 encoder를 CNN의 encoder로 변환하여 이를 구현하였습니다. CNN에서도 충분히 이미지의 정보를 잘 압축하기에 사전에 image classification task로 사전 학습 시킨 후 CNN을 RNN의 encoder로 대체하였습니다. 그리고 해당 모델을 “Neural Image Caption” 이라고 부릅니다.

이러한 방법을 통해서 해당 논문은 3가지 부분에서 기여했다고 주장합니다.

1. 자동 이미지 설명을 end-to-end system으로 구현
2. CNN과 RNN의 결합.
3. SOTA 달성

Model

모델의 경우 RNN의 encoder-decoder 구조를 그대로 가지고 와서, 이미지에 기계번역 encoder의 역할을 하도록 하였습니다. 그 후 이미지의 특징이 주어진 경우, 단어를 생성시 해당 우도가 최대화가 되도록 모델을 설계하였습니다.

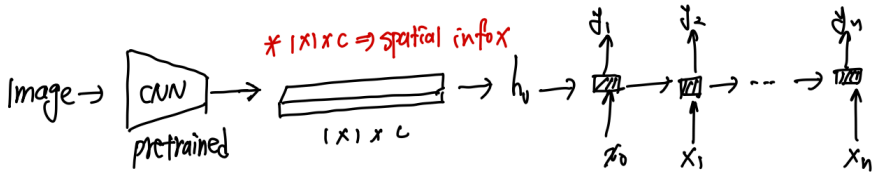
$$\theta^* = \arg\max_\theta \sum_{(I,S)} \log p(S \mid I; \theta)$$

그 후 문장의 길이에는 제한을 두지 않았고, 체인룰에 의해서 모델의 결합 확률을 활용하여 gradient descent를 활용하여 모델을 학습하였습니다.

$$\log p(S \mid I) = \sum_{t=0}^{N} \log p\left(S_t \mid I, S_0, \ldots, S_{t-1}\right)$$

RNN 모델의 경우 LSTM을 활용하였고, CNN의 경우 가장 성능이 좋은 사전 확습된 모델을 활용하였습니다. (f = LSTM)

$$h_{t+1} = f\left(h_t, x_t\right)$$



나의 생각 :

현재 이미지를 단순히 1 x 1 x C 로 압축하기에 이미지의 공간정보를 잃을 수 있기에, 공간정보를 담을 수 있는 방법을 제안해야한다.

LSTM-based Sentence Generator

기존의 RNN 모델들의 경우 학습 과정의 backprogation 시 기울기의 소실, 폭발이 매우 도전적인 문제였습니다. 하지만 LSTM의 경우 이를 3개의 gate를 활용하여 해결하였습니다.

$$\begin{aligned}
(4) \quad i_t &= \sigma\left(W_{ix} \cdot x_t + W_{im} \cdot m_{t-1}\right) \\
(5) \quad f_t &= \sigma\left(W_{fx} \cdot x_t + W_{fm} \cdot m_{t-1}\right) \\
(6) \quad o_t &= \sigma\left(W_{ox} \cdot x_t + W_{om} \cdot m_{t-1}\right) \\
(7) \quad c_t &= f_t \odot c_{t-1} \; + \; i_t \odot h\left(W_{cx} \cdot x_t + W_{cm} \cdot m_{t-1}\right) \\
(8) \quad m_t &= o_t \odot c_t \\
(9) \quad p_{t+1} &= \mathrm{Softmax}(m_t)
\end{aligned}$$

다음과 같은 식을 따라서 각 게이트(입력, 잊음, 출력 게이트)는 학습 가능한 파라미터를 통해 이전의 중요한 정보를 적절히 유지하거나 제거하며, 이렇게 업데이트된 cell state와 output이 결합되어 최종적으로 softmax를 통해 다음 단어의 확률 분포를 예측하게 됩니다.

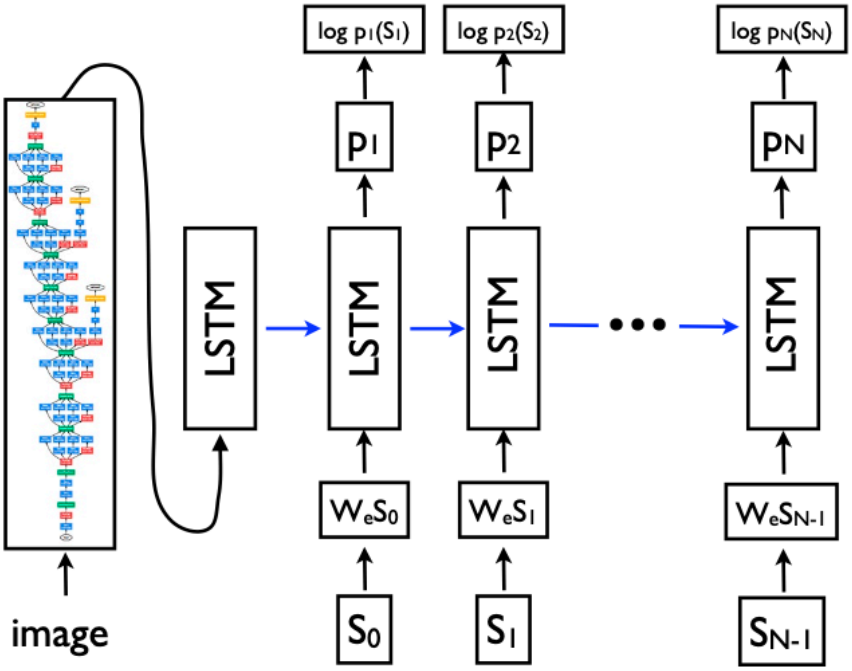


Figure 3. LSTM model combined with a CNN image embedder (as defined in [12]) and word embeddings. The unrolled connections between the LSTM memories are in blue and they correspond to the recurrent connections in Figure 2. All LSTMs share the same parameters.

최종적인 모델 구조의 경우 다음과 같이 CNN의 결과를 Feature Vector로 변환하여 LSTM의 hidden state로 한번 들어가게 됩니다. 이후 이미지의 정보가 담긴 cell state를 통해서 순차적으로 단어를 예측하게 됩니다.

$$
\begin{aligned}
x_{-1} &= \mathrm{CNN}(I) \\
x_t &= W_e S_t, \quad t \in \{0, \ldots, N-1\} \\
p_{t+1} &= \mathrm{LSTM}(x_t), \quad t \in \{0, \ldots, N-1\}
\end{aligned}
$$

다음과 같이 처음 한번에 대해 Image의 feature vector가 입력으로 들어가게 되고, 이후에는 각각의 단어가 순차적으로 들어가게 됩니다. 그리고 각각의 단어는 원핫 인코딩 되어 들어가게 됩니다. 이후 LSTM 을 통과하여 각 확률 분포를 추출하게 됩니다.

$$
L(I, S) = -\sum_{t=1}^{N} \log p_t(S_t)
$$

각 단계별로 $logp_i(S_i)$ 의 값들을 출력하게 되고 모든 i에 대해 결과를 $-\sum$ 하여 최종 손실로 사용하게 됩니다.

Inference

그리고 예측을 진행하는 경우에는 2가지 아이디어를 사용하여 BELU 값을 높혔습니다.

1. Sampling : 랜덤값을 추출하여 매 학습 마다 새로운 단어를 에측할 수 있도록 합니다.
2. BeamSearch : 최대 20개의 후보를 설정합니다. 그리고 누적합을 통해서 매 스텝 가장 높은 확률 값을 갖는 문장을 선택하게 됩니다.

실험을 통해서 Beam = 20으로 설정하는 것이 가장 좋았고, Beam = 1을 하는 경우 BELU 값이 2 감소하였다고 합니다.

Evaluation & Result

평가 방법의 경우 사람이 직접 평가를 진행하는 경우와 BELU를 사용하는 평가 방식을 사용하였고, 두 방법이 연관성이 높아 나머지 부분에 대해서는 모두 자동으로 평가하는 방식을 사용하였다고 합니다.

학습을 하는 경우 오버피팅을 방지하기 위해서 CNN은 사전 학습 가중치를 변경하지 않았고, 임베딩 가중치의 경우 따로 초기화를 진행하지 않았다고 합니다. 그리고 추가로 오버피팅을 막기 위해서 Dropout이나 앙상블 기법을 사용하여 BELU 성능을 올렸다고 합니다.

그리고 실험 결과 이미지 설명 데이터셋이 커지면 모델이 더 많은 데이터를 학습하여 성능이 개선됨을 확인하였습니다. 해당 논문에서는 앞으로 이미지 데이터만 주어진 경우, 혹은 문자 데이터만 주어진 비지도 학습으로도 이미지 설명 모델을 개선할 수 있을것이라고 기대하고 있습니다.

한마디 정리

RNN의 encoder, decoder의 매커니즘을 활용하여 encoder를 이미지 endoer로 변경하여 이미지를 설명하는 end-to-end system을 제안하였습니다. 그리고 실제로 이미지 설명 분야에서 높은 성능을 보여주어 Multi Modal learning의 시작을 알렸습니다.

나의 생각

이미지를 처음에 한번만 입력으로 넣는 것은 효율적일 수 있지만 LSTM의 특성상 후반부로 갈수록 이미지에 대한 정보를 잃고 오로지 이전 단어에 대한 의존도가 높아질것 같다고 판단된다. 그래서 주기적으로 혹은 특정 구간마다 이미지에 대한 정보를 계속 주입시켜주는것도 좋은 아이디어가 될것 같다.