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

Authors: Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho,

Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio

Harvard AC295/CS115

Summary by: Eduardo PEYNETTI, Jessica WIJAYA, Rohit BERI, Stuart NEILSON

# Abstract

An “Attention-based Model” that automatically learns to describe the content of the images:

* Two ways to train the model:
  + Stochastically: By maximizing variational lower bound
  + Deterministically: Using standard backpropagation techniques
* Visualization demonstrates the automatic learning of gaze fixing to generate corresponding words
* Use of attention on three datasets to demonstrate state-of-the-art performance

# Introduction

Automatically generating captions of an image is a task very close to the heart of scene understanding:

* One of the primary goals of computer vision
* Requires capability to capture and express relationships of the detected objects in a natural language
* Mimicking remarkable human ability to compress huge amounts of salient visual information into descriptive language

“Attention” is one of the most curious facets of human visual system:

* Allows for salient features to dynamically come to the forefront as needed
* Particularly important when there is lot of clutter

Top-layer representations distill information in image down to most salient objects:

* Leads to loss of information required for richer and more descriptive captions
* Low-level representations can help preserve this information

Two attention-based image caption generators under a common framework:

* Soft deterministic attention
* Hard stochastic attention

# Related Work

Prior to neural networks, two dominant approaches for image captioning:

* First: Generating caption templates which were filled in based on the results of object detections and attribute discovery – Li et al. (2011), Yang et al. (2011), Mitchell et al. (2012), Kulkarni et al. (2013), Elliot & Keller (2013)
* Second: First retrieving similar captioned images for a large dataset and then modifying these retrieved captions to fit the query – Kuznetsova et al. (2012, 2014)
* Involved intermediate generalization step to remove the specifics of a caption that are only relevant to the retrieved image

Neural networks-based approach to image captioning started in 2014:

* Image captioning is well suited to encoder-decoder framework of machine translation (sequence2sequence):
  + Analogous to translating an image to a sentence
* Kiros et al. (2014a & b) first proposed a multimodal log-bilinear model to be followed by a method to allow for ranking and generation
* Mao et al. (2014) took similar approach but replaced FFNN with RNN
* Vinyals et el. (2014) only showed image to RNN in the beginning in contrast to Kiros and Mao who used image at every time step of the output sequence
* Donahue et al. (2014) used LSTM and applied it to videos to generate video captions
* Karpathy & Li (2014) were the first to propose learning joint embedding space for ranking and generation to score sentence and image similarity using R-CNN object detection with bi-directional outputs of RNN
* Fang et al. (2014) proposed a three-step pipeline for generations by incorporating object detections

Long list of prior work incorporating “attention” in computer vision tasks:

* Similar works in spirit include Larochelle & Hinton (2010), Denil et al. (2012), & Tang et al. (2014)
* Directly extents the works of Bahdanau et al. (2014), Mnih et al. (2014), & Ba et al. (2014)

# Image Caption Generation with Attention Mechanism

The proposed “attention” framework does not explicitly use object detection (unlike most prior works):

* Learns latent alignment from scratch
* Allows model to learn to attend to abstract concepts

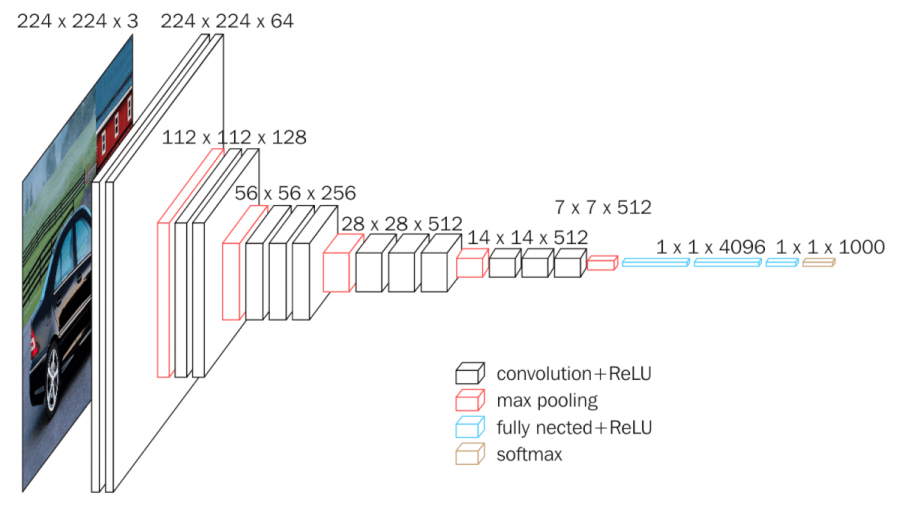
## Model Details

Diagram

Description automatically generated

* Two variants of the attention-based model
  + The main difference is the definition of the φ function (soft and hard attention)
* Input is a single raw image
* Output is the caption y, encoded as a sequence of 1-of-K encoded words:
  + y = {y1, … , yC } , yi ∈ Rk
  + where:
    - k is the size of the vocabulary, and
    - C is the length of the caption

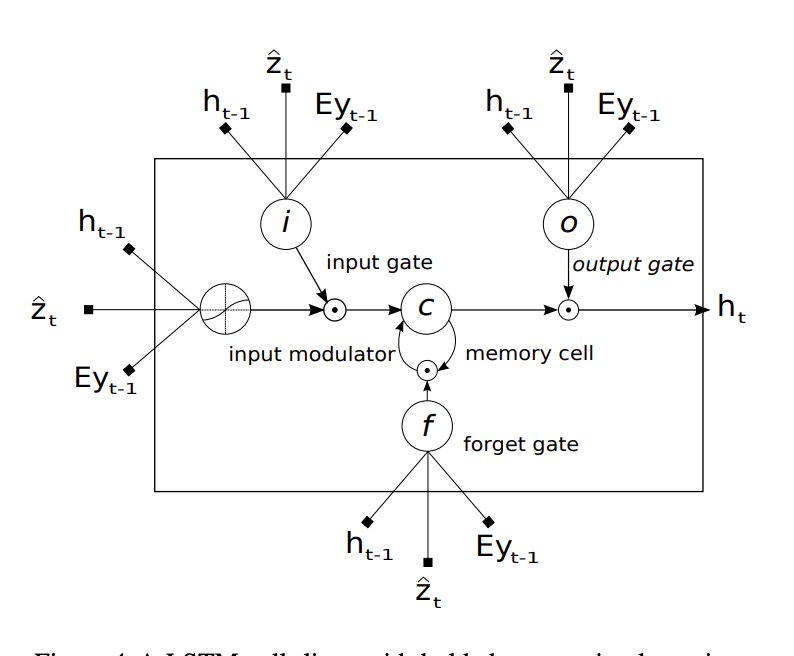
### Encoder: Convolutional Features



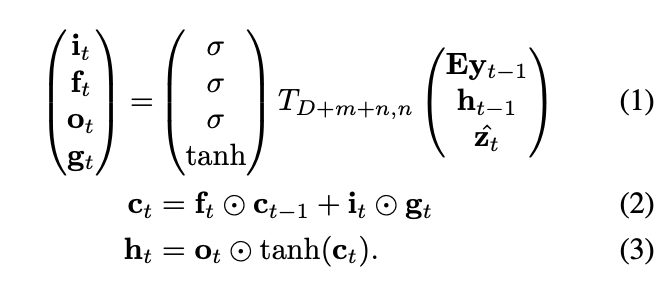


* Uses convolutional neural network to extract a set of feature vectors, i.e., annotation vectors:
  + a = {a1, . . ., aL}, ai ∈ RD
  + L – D-dimensional representation vectors corresponding to a part of the image
* Extract features from a lower convolutional layer
  + To obtain a correspondence between the feature vectors and portions of the 2-D image.
  + Allows the decoder to selectively focus on certain parts of an image by selecting a subset of all the feature vectors.
* Oxford VGGNet was used for “Encoder”
  + Pre-trained on ImageNet
  + Without fine-tuning
  + Used 14 x 14 x 512 feature map of 4th convolutional layer before MaxPooling
  + Flattened into 196 x 512 (i.e. L x D) encoding for Decoder

### Decoder: Long Short-Term Memory



* LSTM network is used to generate one word at every time step, ***conditioned on***:
  + a context vector zt,
  + the previous hidden state ht-1, and
  + the previously generated words yt-1



* + where:
    - it = input state
    - gt = input modulator
    - ft = forget state
    - ct = memory
    - ot = output
    - ht = hidden state
    - zt ∈ RD = context vector, to capture the visual information associated with a particular input location
    - E ∈ Rm × K = embedding matrix of dimension m x K
    - σ and be the logistic sigmoid activation & element-wise multiplication respectively

### Algorithm

* The initial memory state and hidden state of the LSTM are predicted by an average of the annotation vectors

Text

Description automatically generated

* The weight αi of each annotation vector ai is then computed by an attention model fatt for which we use a multilayer perceptron conditioned on the previous hidden state ht−1.

Text

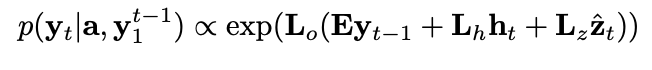
Description automatically generated

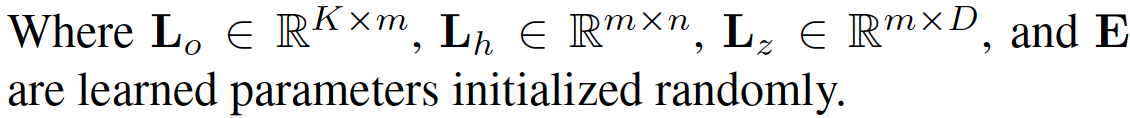
* For each location *i*, the mechanism generates a positive weight αi:
  + In hard attention: αi is the probability that location *i* is the right place to focus for producing the next word
  + In soft attention: αi is the relative importance to give to location *i* in blending the αi’s together.
* The mechanism φ will compute zt from the annotation vectors ai (corresponds to the features extracted at different image locations)

Text

Description automatically generated

* The context vector zt is a dynamic representation of the relevant part of the image input at time *t*
* The hidden state varies as the output RNN advances in its output sequence: “where” the network looks next depends on the sequence of words that has already been generated
* A deep output layer is then used to compute the output word probability given the LSTM hidden state, the context vector and the previous word





# Learning Stochastic “Hard” vs Deterministic “Soft” Attention

A picture containing photo, different, view, display

Description automatically generated

## Stochastic “Hard” Attention

A picture containing window

Description automatically generatedA bunch of different colors

Description automatically generated

* **“Hard Attention”: φ returns a sampled ai at every timestep based on multinouilli distribution**
* st: The location where the model focuses on when generating the tth word
  + st,i: Indicator variable – 1 if the ith location is used to generate the tth word, 0 otherwise
* Attention locations are intermediate latent variables; st and zt being randomly distributed as follows:
  + tth word focus will depend on the location that the previous words have already focused on

Logo, company name

Description automatically generated

A close up of a clock

Description automatically generated

* Objective function Ls is a variational lower bound on marginal log-likelihood *log p(y|a)* of observing the sequences of words *y* given image feature *a*

Text, letter

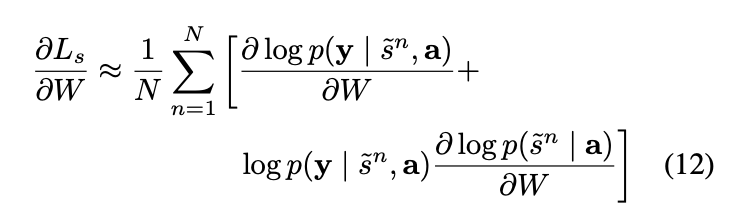
Description automatically generated

* The parameters *W* can be learned by optimizing Ls

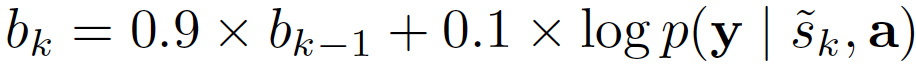
Text, letter

Description automatically generated

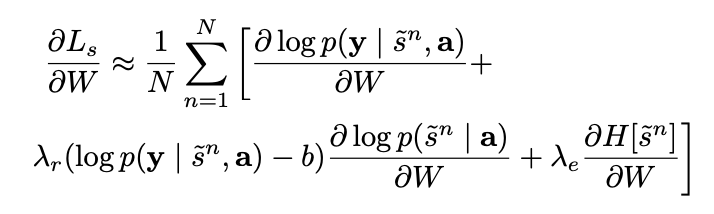
* Parameters (W) are learned by computing gradient using Monte-Carlo based sampling approximations i.e., sampling the location st from the multinouilli distribution



* Sampling based approximations might be problematic when the variance is large (estimation process takes longer to converge and is not efficient). Techniques to reduce variance:
  + Use a moving average baseline (can be estimated as an accumulated sum of the previous log likelihoods with exponential decay)



* + Add an entropy term on the multinouilli distribution (H[s]).
  + Also, with probability 0.5 for a given image, we set the sampled attention location s to its expected value α.
* The final learning rule for the model is equivalent to the REINFORCE Learning Rule – Reward for choosing a sequence of actions is proportional to log-likelihood of the target sentence



* + Where: and are two hyper-parameters set by cross-validation

## Deterministic “Soft” Attention

Graphical user interface, application

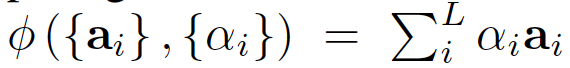
Description automatically generatedA picture containing small, sitting, photo, room

Description automatically generated

* **Unlike “Stochastic Hard Attention” which requires sampling the attention location st each time, in of “Deterministic Soft Attention” we compute the expectation of the context vector directly**

A close up of a clock

Description automatically generated



* Unlike “Stochastic Hard Attention”, this model is differentiable and amenable to standard backpropagation. Model here is optimizing the following equation:

Text, letter

Description automatically generated

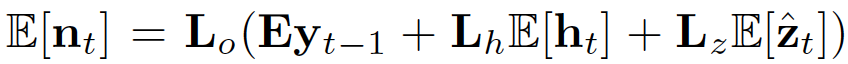
* Hidden activation ht is a linear projection of the context vector zt followed by *tanh* activation.
* Deterministic Attention model approximates the marginal likelihood over the attention locations.
* Normalized Weighted Geometric Mean for the softmax kth word prediction is given by:

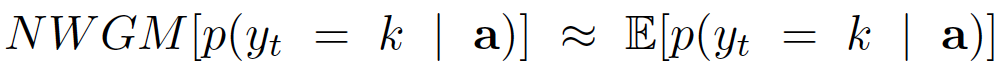
Text, letter

Description automatically generated

* + Where:







### Doubly Stochastic Attention

* By construction we have (softmax):

Text, letter

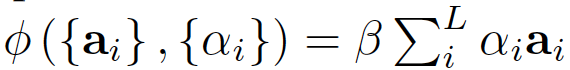
Description automatically generated

* Additional regularization for “Soft Attention”:

Text, letter

Description automatically generated

* Regularization forces algorithm to pay equal attention to every part of the image over the course of caption generation
  + Quantitatively: leads to improvement in BLEU score
  + Qualitatively: leads to more rich and descriptive captions
* Additionally, the “Soft Attention” models predicts a scaler from previous hidden state ht-1



A close up of a clock

Description automatically generated

* + leads to attention weights being emphasized on objects in the images
* Model minimizes the penalized negative log-likelihood

A close up of a watch

Description automatically generated

## Training Procedure

* Attention models (soft and hard) were trained with SGD using adaptive learning rate
  + Flickr8k dataset: RMSProp worked best
  + Flickr30k/MS COCO datasets: ADAM optimizer was used
* Implementation requires time proportional to the length of the longest sentence per update
  + Hence training on random group of captions of different sizes were computationally wasteful
  + Formed mini batches of size 64 with randomly selected equal length captions
    - Greatly improved convergence speed
  + MS COCO with soft attention took 3 days to train on NVIDIA Titan Black GPU
* Regularization
  + Dropout and early stopping on BLEU score was used
  + Breakdown in correlation between validation set log-likelihood and BLEU in later stages of training
  + Used BLEU for model selection
* Whetlab was used for hyperparameter tuning for Flickr8k using soft attention
  + Insights were useful for other datasets as well
* Theano was used for coding

# Experiments

5 different architectures are compared across 3 different datasets with 2 different performance measures. In total 63 different comparisons are made.

## Data

* Three different datasets are used in experiments:
  + Flickr8k – 8,000 images – 5 reference sentences per image
  + Flickr30k – 30,000 images – 5 reference sentences per image
  + Microsoft COCO – 82,783 images – some images have more than 5 reference sentences
* Basic tokenization was used for all three datasets for consistency
* Fixed vocabulary size of 10,000 words was used

## Evaluation Procedures

* Comparison is made against architectures which use GoogleNet or Oxford VGG
  + Google NIC (Vinyals et al., 2014)
  + Log Bilinear (Kiros et al., 2014a)
* Some additional models using AlexNet are also compared with METEOR
  + CMU/MS Research (Chen & Zitnick, 2014)
  + MS Research (Fang et al., 2014)
  + BRNN (Karpathy & Li, 2014)
* Ensembling is not used
* Used predefined splits or publicly available splits

## Quantitative Analysis

* State of the art performance on all three datasets without using ensemble
* Big Boost in METEOR performance is likely due to:
  + Regularization techniques
  + Use of lower-level representation

### BLEU



* Bilingual Evaluation Understudy (BLEU) is an algorithm for [evaluating](https://en.wikipedia.org/wiki/Evaluation_of_machine_translation) the quality of machine translation
* Ranges between 0 and 1 or 0% and 100%
* Uses a modified form of precision to compare a translation against multiple reference translations
* Frequently been reported as correlating well with human judgement
  + However, number of criticisms have been voiced

### METEOR

* Metric for Evaluation of Translation with Explicit Ordering (METEOR) is a metric for the evaluation of machine translation output
* Based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision
* Produces good correlation with human judgement at the sentence or segment level
  + Results have been presented which give correlation of up to 0.964 with human judgement at the corpus level, compared to BLEU's achievement of 0.817
  + At the sentence level, the maximum correlation with human judgement achieved was 0.403

Table

Description automatically generated

## Qualitative Analysis: Learning to attend

* Visualizing the attention component adds an extra layer of interpretability
* Model learns alignment that corresponds to human intuition
* Possible to exploit visualizations to get an intuition of the reasons for mistakes by algorithm

Graphical user interface, website

Description automatically generated

# Conclusion

“Attention-based” approach given state-of-the-art performance on the three datasets.

* Learned “attention” can be exploited to give more interpretation for the model’s generation process
* Learned alignments correspond very well to human intuition

Bonus Article:

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

Authors: Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn,

Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,

Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby

Harvard AC295/CS115

Summary by: Eduardo PEYNETTI, Jessica WIJAYA, Rohit BERI, Stuart NEILSON

For Natural Language Processing, we have seen the evolution RNN -> RNN with Attention -> Transformer Only

For image processing, the previous article similarly shows the progression from CNN -> CNN with Attention -> Can we also go to Transformer only for images?

----

This new article shows that the answer is yes.

A conference paper for the 2021 ICLR conference, published on ArXiv last week (an earlier version in which the author names were redacted for blind review was up a few weeks earlier).

It introduces the ViT (Vision Transformer) model.

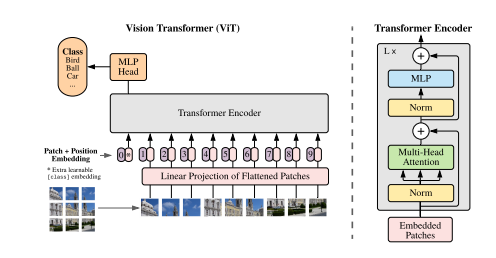
-----

Challenges with using Attention for images:

* Attention relates every pixel to every other pixel
  + Computational complexity O(n2) where n is the number of pixels (n4 if you think of n as the image width)
  + Using Convolutional layers reduced this to a reasonable size by making a more compact latent representation

How this article handles this challenge: divide the image into “patches” of 16x16 = 256 pixels

-------



-------

The patches are flattened, then combined with a “position embedding” which is trainable (you can also see an extra zero position component on the left-hand side – which is analogous to the “cls” tag in BERT).

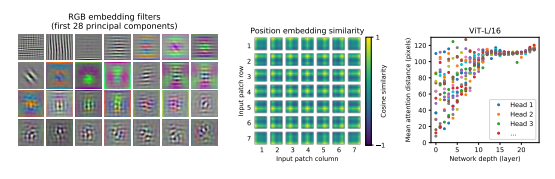
This is then passed into a standard “off-the-shelf" transformer.

Then finally, it goes through a Feed Forward Network to get to a classification prediction (the authors use the terminology MLP – Multi-Layer Perceptron).

This architecture can take advantage of parallelization in the calculations.

-------

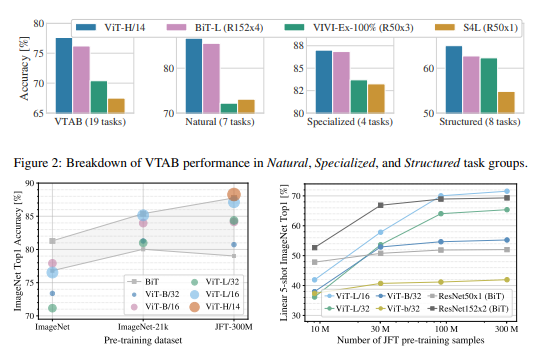
Visualization of their embedding filters indicates that they exhibit a learning process quite similar to Convolutional filters.



--------

The model achieves state-of-the-art performance in image classification

(it is the blue bars)



------

Question:

* Can such an approach be extended from image classification to caption generation as was done in the previous article?
  + Recall that the previous article fully reruns its attention process after each word in the generated caption – would a transformer for image captioning also want to do the same?

# Abstract

# Introduction

# Related Work

# Method

## Vision Transformer (TF)

### Hybrid Architecture

## Fine-Tuning and Higher Resolution

# Experiments

## Setup

### Datasets

### Model Variants

### Training & Fine-Tuning

### Metrics

## Comparison to State of the Art

## Pre-Training Data Requirements

## Scaling Study

## Inspecting Vision Transformer

## Self-Supervision

# Conclusion