Attention Based Image Captioning

Nitesh Saurav
Net ID: NS3664

Saumya Mohan
Net ID: SM7105

1. Introduction

Recognition and description of images and videos is a fundamental challenge of computer vision. Dramatic progress has recently been achieved by supervised Convolutional Neural Network (CNN) models on image recognition tasks. Automatically describing the content of an image using grammatically correct english sentences is a much more challenging task than image classification or object recognition tasks. It, however, has far reaching impact such as helping visually impaired people better understand the content of images on the web. The challenging part of this task is due to the fact that the generated description must not only capture the objects contained in an image, but must also express their attributes, the manner in which these objects relate to each other, and the activities these objects are involved in. Additionally, the above semantic knowledge needs to be expressed in a natural language such as English, thus requiring a language model in addition to visual understanding. In this paper, we have experimented with a model based on Long-Short Term Memory (LSTM) and deterministic attention to caption images.

2. Related Work

Most of the recently proposed methods for generating image captions are based on Recurrent Neural Networks (RNN) and are inspired by the use of sequence to sequence training with neural networks for machine translation.

Some of the neural-network based models that have be proposed in the past represent images as a single feature vector from the top layer of a pre-trained Convolutional Neural Network (CNN). One such early model, proposed by Kiros et al.[1], included a multimodal log-bilinear model biased by image features, on which an improvement was later made to explicitly allow a natural way of ranking and regression. Another such model, proposed by Mao et al., deviated from the first model by using a recurrent neural network instead of feed-forward neural language model, while yet another, proposed by Donahue (2014) [2], used LSTM RNNs and were later extended to generate descriptions for videos as well. These models were designed to see the image at each time step of the output word sequence. There have also been models, such as those proposed by Vinyals et al., in which the RNN sees the image only at the beginning.

Another class of models, proposed by Karpathy and Li (2014) [3], included learning a joint embedding space for ranking and generation where the model learned scoring of sentences and image similarities as a function of R-CNN object detections and created a bidirectional RNN as output. An enhancement on this model, proposed by Fang et al. (2014) [4], included a three-step pipeline for generation with object detection. In these models, the detectors for visual concepts are first learned based on a multi-instance learning framework. Then, a language model trained on captions is applied to the detector output, and finally rescoring from a joint image-text embedding space is done to generate image captions.

Recent research work in this domain heavily involves study of attention based neural encoder-decoder models, where attention mechanism typically produces a spatial map highlighting image regions relevant to each generated word. This in turn helps to incorporate fine-grained visual clues from the images allowing meaningful captioning of the images.

3. Method

The input image is given to a Convolutional Neural Network (CNN) which can be thought of as an encoder to extract the features commonly known as annotation vectors. The last hidden state of the CNN is connected to the rest of the model. This allows the decoder to selectively focus on certain parts of an image by selecting a subset of all feature vectors. In

our model, we use pre trained Resnet 152 for this (can be found here). On the decoder part, our model uses a neural and probabilistic framework which uses LSTM with deterministic soft attention to output descriptions from images, generating one word at every time step conditioned on a context vector, the previous hidden state and the previously generated words

Our model contains the following components: Pre-trained CNN, words embeddings, and LSTM with soft attention.

General Description

The probability of the correct description given the image is directly maximized by using the equation:

$$\theta^* = argmax_\theta \sum_{(I,S)} \log p(S \mid I; \theta) \qquad (1)$$

Here, θ are the model parameters, I is an image, and S is that image's correct transcription. S represents any sentence and so its length is unbounded and chain rule can be applied to model the joint probability over $S_0, \ldots, S_N$, with length N as

$$\log p(S \mid I) = \sum_{i=0}^{N} \log p(S_t \mid I, S_0, \ldots, S_{t-1}) \qquad (2)$$

(S, I) is a training pair example. Sum of the log probabilities, as shown in equation (2), is optimized over the whole training set using stochastic gradient descent. $p(S_t \mid I, S_0, \ldots, S_{t-1})$ is modeled with an RNN, where the variable number of words conditioned upon till $t-1$ is expressed by a fixed length hidden state or memory h_t. This memory is updated after seeing a new input x_t by using a non-linear function f, which can deal with vanishing and exploding gradients.

$$h_{t+1} = f(h_t, x_t) \qquad (3)$$

Equation (3) is modeled using an LSTM network.

Working with LSTM

LSTM is a special kind of RNN capable of learning long-term dependencies. Its chain-like structure has a repeating module that has four neural networks. LSTM's core is a memory cell c (shown in Figure 1 (a)) which runs down the entire chain and keeps knowledge at each time step of which inputs have been observed so far. Information can be added or removed from the memory cell and is regulated by three structures called gates. Gates are composed of a sigmoid neural network layer and a point wise multiplication operation.

The sigmoid layer $\sigma(.)$ outputs numbers between zero and one, describing how much of each component should be let through, value of one meaning to let everything through. Then, the tanh layer $h(.)$ creates a vector of new candidates c_t that could be added to the state. In the next, we combine these two to create an update to the state.

The gates, cell update, and output are defined as:

input gate layer $i_t = \sigma(W_{ix} x_t + W_{im} m_{t-1}) \qquad (4)$

forget gate layer $f_t = \sigma(W_{fx} x_t + W_{fm} m_{t-1}) \qquad (5)$

output gate. $o_t = \sigma(W_{ox} x_t + W_{om} m_{t-1}) \qquad (6)$

$$c_t = f_t \odot c_{t-1} + i_t \odot h(W_{cx} x_t + W_{cm} m_{t-1}) \qquad (7)$$

$$m_t = o_t \odot c_t \qquad (8)$$

$$p_{t+1} = \text{Softmax}\,(m_t) \qquad (9)$$

Here, $\odot$ represents point-wise multiplication with a gate value, and the W matrices are the trained parameters. In equation (9), softmax can be thought of the max of the relevance of the variables. p_{t+1} has only one value as 1, rest are 0.

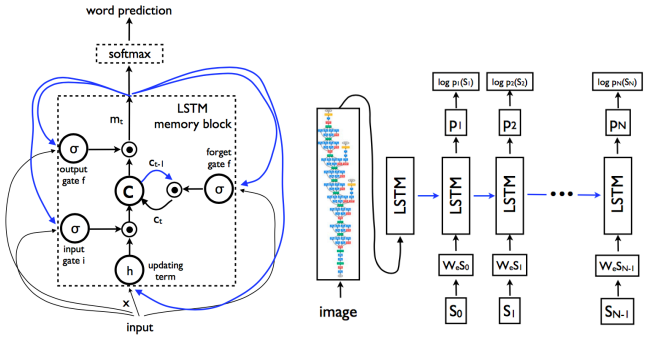


Figure 1: (a) (Left) LSTM, recurrent connections in blue; (b) (Right) LSTM model combined with CNN image embedder and word embeddings

The LSTM model is trained to predict each word of the sentence after it has seen the image and all preceding words defined by $p(S_t \mid I, S_0, \ldots, S_{t-1})$. A copy of the LSTM memory is created for the image and each sentence word such that all LSTMs share the same parameters, and the output m_{t-1} of the

LSTM at time $t-1$ is fed to the LSTM at time t (Figure 1(b)). All recurrent connections are transformed to feed-forward connections in the unrolled version. The unrolling procedure is:

$$x_{-1} = CNN(I) \quad (10)$$

$$x_t = W_e S_t,\ t \in \{0...N-1\} \quad (11)$$

$$p_{t+1} = LSTM(x_t),\ t \in \{0...N-1\} \quad (12)$$

where each word is represented as a one-hot vector S_t of dimension equal to the dictionary size. By emitting the stop word, the LSTM signals that a complete sentence has been generated. Both the image and the words are mapped to the same space, the image by using a vision CNN, the words by using word embedding W_e. The image I is only input once, at $t=-1$, to inform the LSTM about the image contents.

The loss $L(I,S)$ is the sum of the negative log likelihood of the correct word at each step.

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

The above loss is minimized w.r.t. all the parameters of the LSTM, the top layer of the image embedder CNN and word embeddings W_e.

BeamSearch approach is used in our model to generate a sentence, given an image. This approach iteratively considers the set of the k best sentences up to time t as candidates to generate sentences of size $t+1$, and keep only the resulting best k of them. This gives a good approximation for $S = argmax_{s'p}(S'|I)$.

Deterministic Soft Attention Model used in LSTM

An attention model is a method that takes n arguments $y_1, \ldots, y_n$ and a context c and returns a vector z which is the summary of the y_i focussing on information linked to the context c. More formally, it returns a weighted arithmetic mean of the y_i, and the weights are chosen according to the relevance of each y_i, given the context c.

For our model, the context c is the beginning of the generated sentence, y_i are the annotation vectors of the image given by the CNN (encoder), and the output is a representation of the filtered image, with a filter putting the focus on the interesting part for the word currently generated.

The softmax calculated in equation (9) and CNN's annotation vectors are used to get the output z.

The output z is the weighted arithmetic mean of all the annotation vectors y_i, where the weight represent the relevance for each variable according to the corresponding index of softmax output.

The whole model is smooth and differentiable under the deterministic attention, so learning end-to-end is trivial by using standard back propagation.

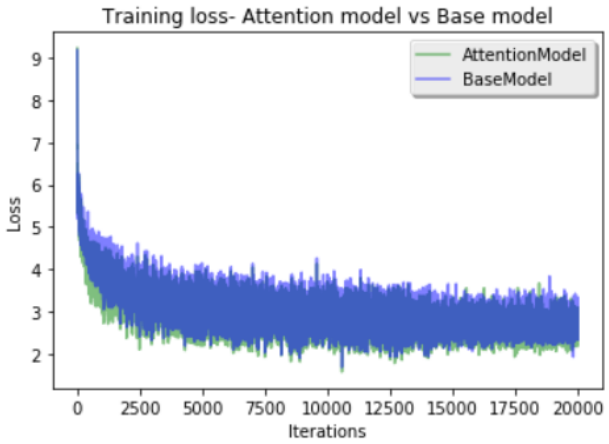
4. Experiments & Results

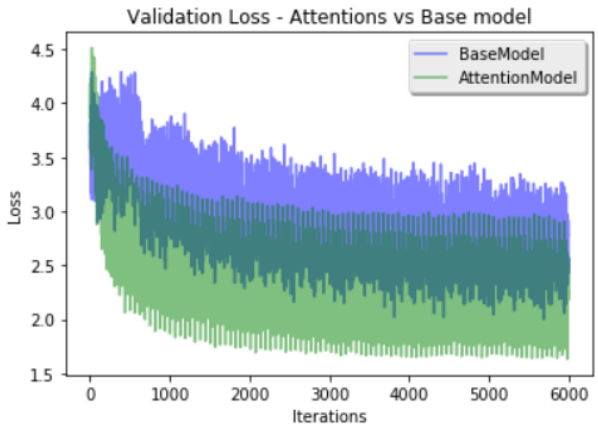
We implemented an attention based Image Captioning model enunciated in the paper [5], and evaluated the improvements over the baseline model based on the paper Show and Tell by Vinyals [6]. The model were trained on MS COCO dataset and evaluated on BLEU scores and CIDER. To further improve our model we experimented with the optimization function, RNN size, RNN types etc.

Following are graphs from each variant we tried.

4.1 Results with Attention model v/s baseline model

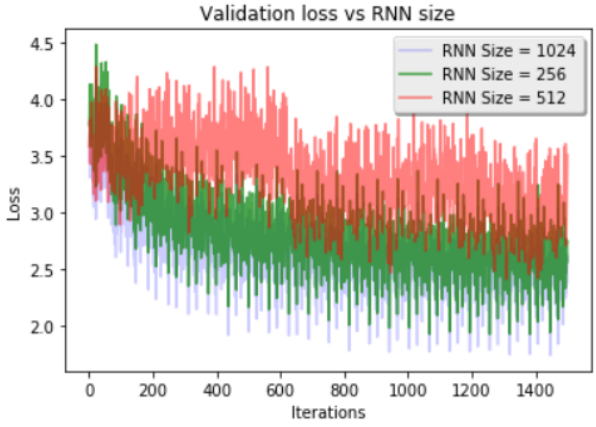
Our model with attention was able to achieve better train and validation loss compared to base model. We also evaluated the model for generated captions shown in below section 4.5





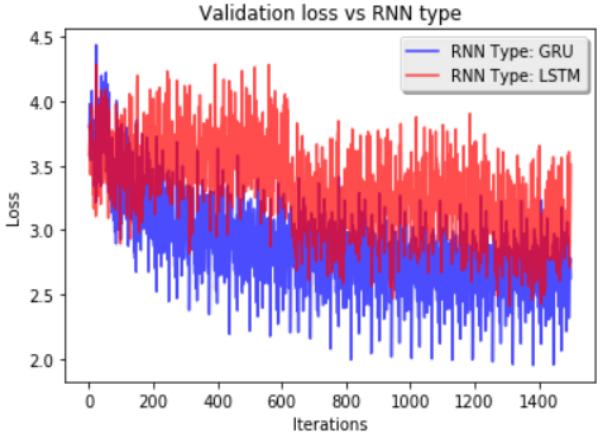
4.2 Results with various RNN hidden size

The hidden layer size is an important hyper parameter affecting the LSTM network performance. The larger networks perform better, and the required training time increases with the network size.

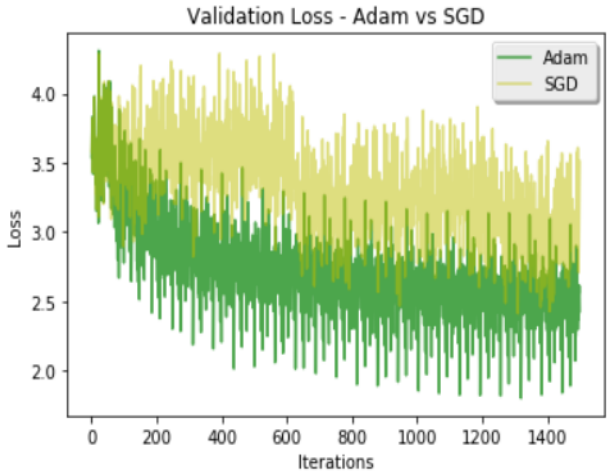


4.3 Results with different variants of RNN

Training was faster with GRU as compared to LSTM since GRUs are computationally more efficient. Also, for our model GRU performed better than LSTM.



4.4 Adam vs SGD



In our experiments with Adam and SGD, better validation performance was achieved using the Adam optimizer.

4.5 Sample Results – Generated captions for images

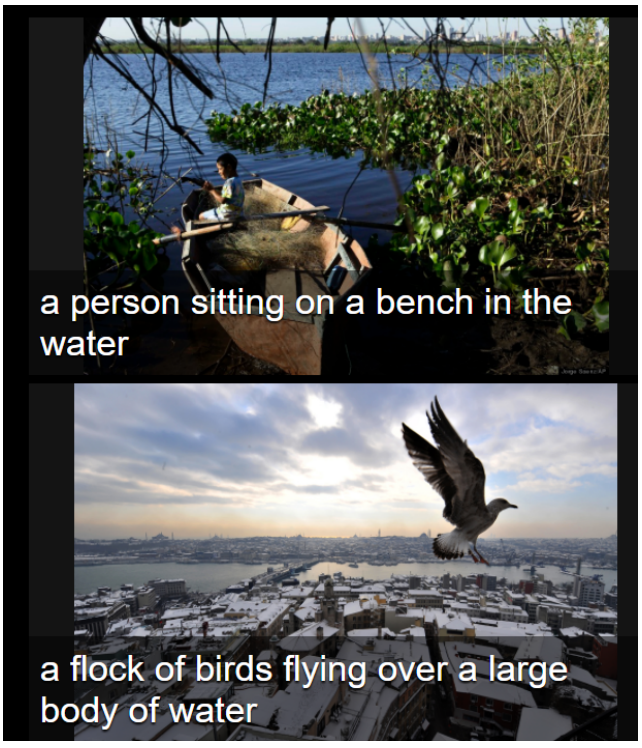


Figure 2: captions with base model

Figure 3: captions with Attention model

We can clearly see that attention based models generate more accurate captions.

4.6 Test on raw images

We captioned an image which was not present in the training set. The resultant caption generated was “a brown and white dog is standing in a field”. We observed that as the earlier training images had a lot of images with dogs in it, the model is biased in captioning a lion with a dog. Although, other information like field and color of object regarding the image is correct. Thus, we can conclude that sometimes the captions are wrongly generated owing to bias in the dataset.

4.7 Official Scores - Bleu score,CIDEr-D

Model	BLEU Score	CIDEr-D
Base Model	0.692	0.83
Our Model	0.725	0.95

5. Conclusion

We read the existing literature on image captioning and chose Show Tell [6] code as our baseline model. We improved it by using attention as described in Show Attend and Tell [5]. To further improve the model, we experimented with various optimization functions and RNN size. Our results in section 4 show that attention based model gave better Bleu and CIEr-D scores and generated more accurate captions for images as compared to the baseline model.

6. References

1. Kiros, Ryan, Salakhutdinov, Ruslan, and Zemel, Richard. Uni- fying visual-semantic embeddings with multimodal neural lan- guage models. arXiv:1411.2539, 2014

2. Donahue, Jeff, Hendrikcs, Lisa Anne, Guadarrama, Se- gio, Rohrbach, Marcus, Venugopalan, Subhashini, Saenko, Kate, and Darrell, Trevor. Long-term recurrent convo- lutional networks for visual recognition and description. arXiv:1411.4389v2, 2014

3. A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. NIPS, 2014.

4. Fang, Hao, Gupta, Saurabh, Iandola, Forrest, Srivastava, Rupesh, Deng, Li, Dolla´ r, Piotr, Gao, Jianfeng, He, Xiaodong, Mitchell, Margaret, Platt, John, et al. From captions to visual concepts and back. arXiv: 1411.4952, 2014

5. Xu, K. et al. Show, attend and tell: Neural image caption generation with visual attention. In Proc. International Conference on Learning Representations arxiv:1502.03044, 2015.

6. Vinyals, O., Toshev, A., Bengio, S. & Erhan, D. Show and tell: a neural image caption generator. In Proc. International Conference on Machine Learning arxiv: 1502.03044, 2014