Image Caption Generator using LSTM and Bidirectional LSTM

Abhinav Tiwari | Ronak Mistry | Shivam Negi
CSYE 7245, Spring 2018, Northeastern University

Abstract

This paper aims to compare the performance of Long short-term memory (LSTM) networks with Bidirectional long short-term memory (BLSTM) networks using a task of caption generation as the medium. A BLSTM is more complex compared to a conventional LSTM and has a greater number of training parameters. It is also computationally expensive. The common consensus in the industry is that this implies a gain in performance.

The paper aims to quantify this performance gain for BLSTM networks. We have used the Flickr8k dataset along with transfer learning principles to train pretrained image classification models to extract features from an image and map them to user defined captions. The pretrained models used are InceptionV3 and MobileNet. Both of these models are trained against the ImageNet dataset and MobileNet is the lighter of the two.

The models are then evaluated using BLEU(bilingual evaluation understudy) scores to assess the accuracy of captions generated for images.

Introduction

The technology behind computer vision based image caption generation models have made considerable progress in recent years. ImageNet visual recognition challenges have brought several researchers and institutions together to develop algorithms and research in the field of object category classification and detection on hundreds of object categories and millions of images. Developments in this area finds its use in several applications like generating applications for visually impaired people, autonomous vehicles and categorizing images based on labels.

Modern day research in this field is being spearheaded by Google Brain team which proposed Show and Tell: A Neural Image Caption Generator. The paper presented by google replaced the encoder RNN by a deep convolution neural network (CNN). As CNN can be leveraged to produce a source for the model by embedding the image into a fixed length vector which can be later taken as an input for many other computer vision tasks and image based model. While Convolutional

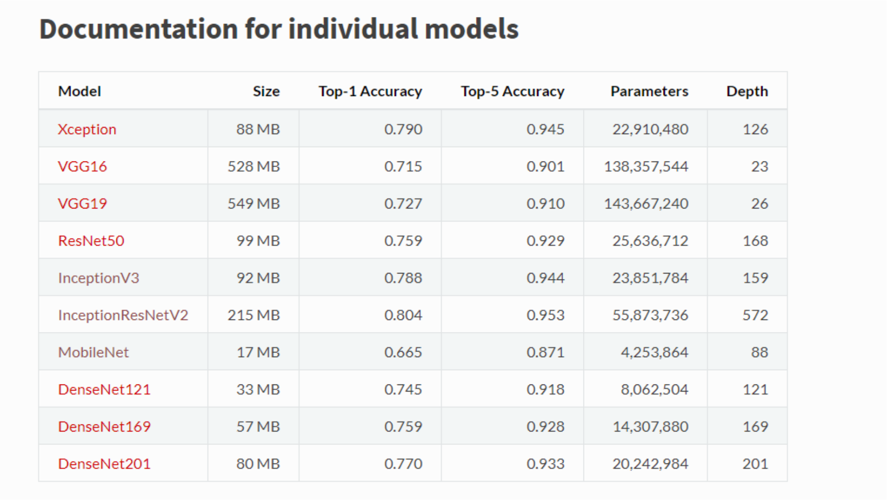
Networks can be broadly used for classification, localization and detection, feature extraction is the key to make an image captioning model. The Convolution Net is trained on image data for image classification task while the hidden layer acts as a source for the input to the Recurrent Neural Network (decoder), which generate a simple sentence describing the image.

This paper is inspired by and based off the work done in the 'Show and Tell' paper by Google. The paper has two major components. The first is using transfer learning to adapt a classification model and perform feature extraction on images. The final candidate models for this were InceptionV3 and MobileNet. The Flickr8k dataset was our choice as a source of images and their associated captions.

For both the models, a recurrent neural network encodes the variable length input into a fixed dimensional vector, which is taken as the maximum length of the caption available mapped with the image and uses this representation to "decode" it to the desired output sentence.

Flickr-8K:

Flickr-8K is a dataset with 8000 images from the flickr website and can be found here. There are 6000 training images, 1000 validation images and 1000 testing images. Each image has 5 captions describing it. These captions act as labels for the images. There is no class information for the objects contained within an image.



Documentation for individual models

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.715	0.901	138,357,544	23
VGG19	549 MB	0.727	0.910	143,667,240	26
ResNet50	99 MB	0.759	0.929	25,636,712	168
InceptionV3	92 MB	0.788	0.944	23,851,784	159
InceptionResNetV2	215 MB	0.804	0.953	55,873,736	572
MobileNet	17 MB	0.665	0.871	4,253,864	88
DenseNet121	33 MB	0.745	0.918	8,062,504	121
DenseNet169	57 MB	0.759	0.928	14,307,880	169
DenseNet201	80 MB	0.770	0.933	20,242,984	201

Figure 1. List of image classification models and their parameters

InceptionV3

In the field of computer vision research, the ImageNet Project is aimed at labeling and categorizing images into almost 22,000 object categories. 1.2 million training images are used to build the model while another 50,000 images for validation and 100,000 images for testing.

The Inception V3 model proposed by Szegedy et al. has a CNN based architecture and led to a new state of the art for classification and detection. It was designed to tackle the ImageNet dataset. The key feature of the model is its design which improved utilization of the computing resources. The design achieves this by allowing for increased depth and width of the model. The weights for Inception V3 are relatively small, with the total size coming in at 96MB.

Architecture

The Inception module is designed as a "multi-level feature extractor" which is implemented by computing 1×1, 3×3, and 5×5 convolutions within the same module of the network. The network is built in such a way that the result obtained from the convolutions is being stacked along the channel dimension and then fed into layer in the network. Figure 2 depicts the architecture of the InceptionV3 model.

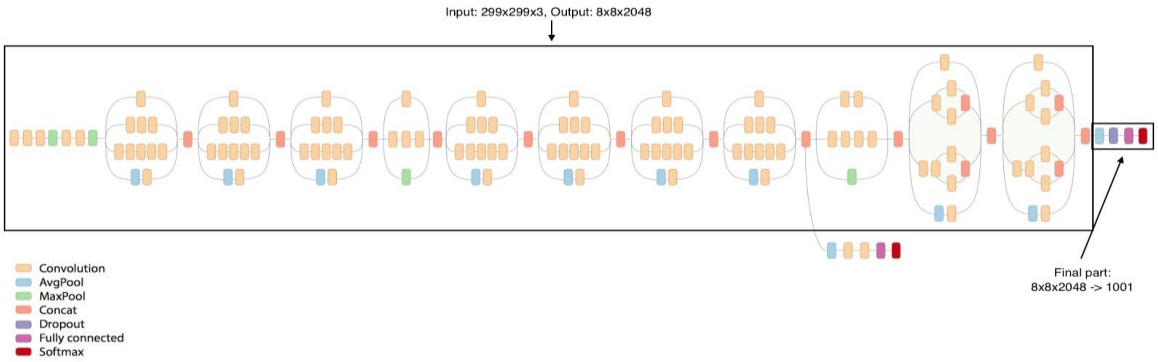


Figure 2. Architecture of the InceptionV3 Model

MobileNet

Computation efficacy is a key factor which drives deep learning algorithms. In 2017, Google's Mobile Net came out as a model which can effectively maximize the accuracy while keeping a tab on resource usage of the device it is run on. The design of MobileNets are built for classification, image segmentation, detection and embedding, and work the same way as other ImageNet models work, however, MobileNets are designed to have a small size, low latency and low power consumption. Figure 3 details the architecture for the MobileNet model.

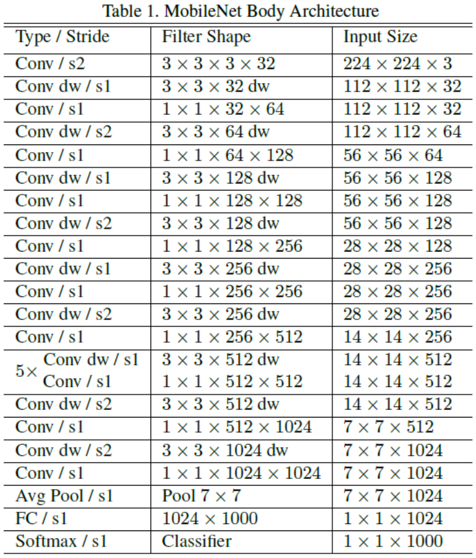


Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Figure 3. Architecture of the MobileNet model

For the second component, our model leveraged LSTM's and Bidirectional LSTM's and compared their performance.

LSTM:

Long short-term memory (LSTM) units are a building unit for layers of a recurrent neural network (RNN). A RNN composed of LSTM units is often called an LSTM network. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell is responsible for "remembering" values over arbitrary time intervals; hence the word "memory" in LSTM. Each of the three gates can be thought of as a "conventional" artificial neuron, as in a multi-layer (or feedforward) neural network: that is, they compute an activation (using an activation function) of a weighted sum. Intuitively, they can be thought as regulators of the flow of values that goes through the connections of the LSTM; hence the denotation "gate". There are connections between these gates and the cell. This is detailed in figure 4.

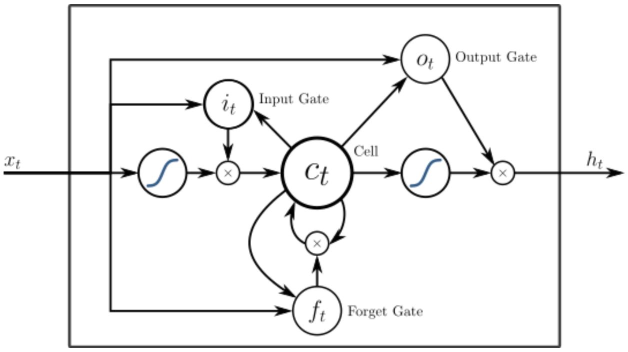


Figure 4. A simple peephole LSTM

The expression long short-term refers to the fact that LSTM is a model for the short-term memory which can last for a long period of time. An LSTM is well-suited to classify, process and predict time series given time lags of unknown size and duration between important events. LSTMs were developed to deal with the exploding and vanishing gradient problem when training traditional RNNs. Relative insensitivity to gap length gives an advantage to LSTM over alternative RNNs, hidden Markov models and other sequence learning methods in numerous applications.

Bidirectional LSTM:

For a B-LSTM, the inputs will run two ways by duplicating the first recurrent layer in the network so that there are now two layers side-by-side. The input sequence is provided as an input to the first layer and the reversed copy of that is provided to the second layer. Though this drastically increases the number of parameters, it also increases the accuracy of the model.

Approach:

Our approach to deploying these components is summarized in figure 5.

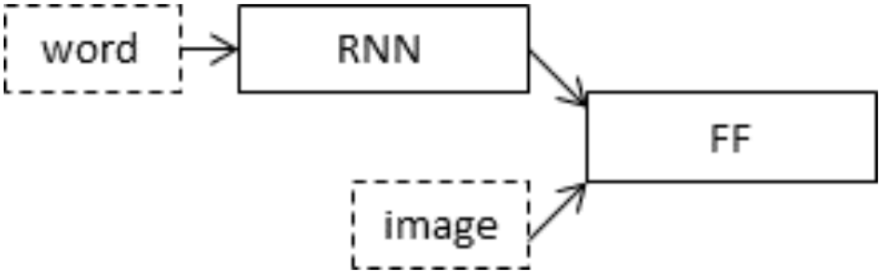


Figure 5. Model architecture

Data Preprocessing:

Preprocessing the data for the models is the first step. The data for flickr8k is divided into two folders. One folder with images and one with captions. The first step is to map these to each other. Using the given token file, a dictionary is created with the images as keys and their value is a set of 5 captions.

Beyond this, images must be scaled to fit the model's input requirements prior to feature extraction.

- Inception-v3 requires the input images to be in a shape of 299 x 299 x 3.

- The MobileNet model requires images to be in a shape of 224 x 224 x 3.

Building a model:

Once the preprocessing is complete, data is fed to 2 parallel components in the architecture. The diagram depicts the 2 parallel phases of the model. The processed input images are passed through either the InceptionV3 or the MobileNet model. This step is to extract features from the image. The final dense layers with 'softmax' functions are dropped from both the models since we do not intend to classify the images. The output from the models is in the form of a flattened array. The InceptionV3 model gives the output features with shape (1, 2048). The MobileNet model provides the same features in the shape of (1, 1024).

Parallelly, the caption data for the training images is fed to the RNN (LSTM/BLSTM). This model takes data from the captions with respect to the words used and their frequency.

The output from both these parallel models is fed to a decoder. The decoder then maps image features to information extracted by the LSTM/BLSTM. Thus, features in images are associated to words in captions to train the model.

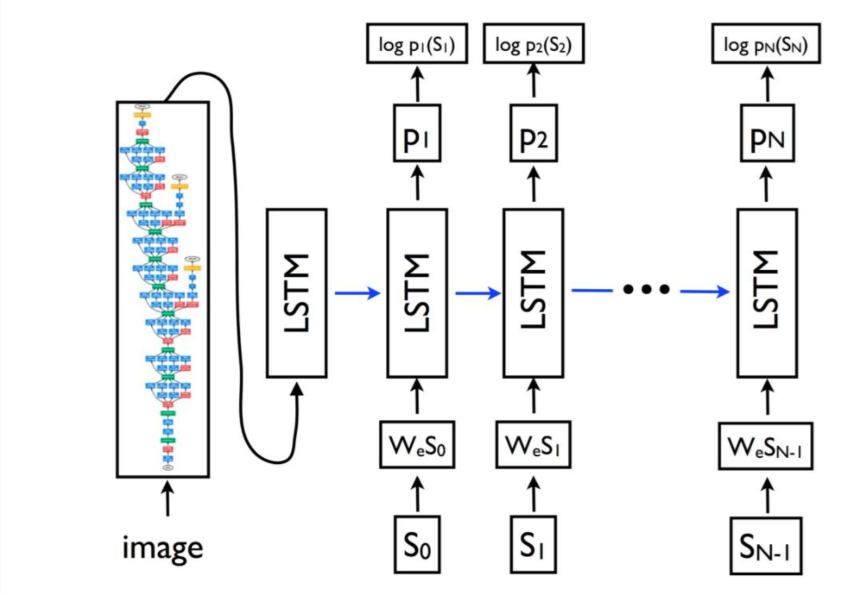


Figure 6. Image feature being mapped to the LSTM Network

Code:

The code and assets for this project can be found at this GitHub repository. It also contains instructions on how to setup an environment and execute the notebook.

Evaluation Metric:

As part of this project, we’ve narrowed down two evaluation metrics. One qualitative and one quantitative.

i. Qualitative

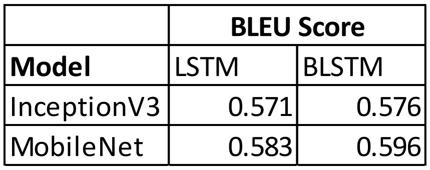
We looked at model performance as one of the factors. This includes the size of the network, the rate at which the network learns, how early it plateaus and how resource intensive the model is.

ii. Quantitative:

To quantifiably measure the accuracy of the models, we will be using the BLEU (bilingual evaluation understudy) score. BLEU was originally developed to assess performance with language translation problems. The primary task for a BLEU implementer is to compare n-grams of the candidate with the n-grams of the reference translation and count the number of matches. These matches are position-independent. The more the matches, the better the candidate translation is. A perfect match would give a score of 1.0 and a perfect mismatch would return a score of 0.0.

The Flickr-8k dataset after preprocessing, provides data in the form of a dictionary where, the key is an image and the value for that image is a set of 5 captions. The BLEU metric is used to compare the predicted caption from the model to all the given caption labels. The best match from all these labels is picked as the final BLEU score.

Results:



	BLEU Score	
Model	LSTM	BLSTM
InceptionV3	0.571	0.576
MobileNet	0.583	0.596

Figure 7. Table detailing BLEU scores for all architectures

The results from all four models are detailed in Figure 7. The final BLEU evaluation was performed against 1000 images. Each image had a predicted caption that was assigned a BLEU score. We have rated each model based on a cumulative average of these scores.

Figures 8 depicts the distribution of BLEU scores over the entire dataset for the InceptionV3 LSTM model. More than 50% of the dataset is classified over a score of 0.5.

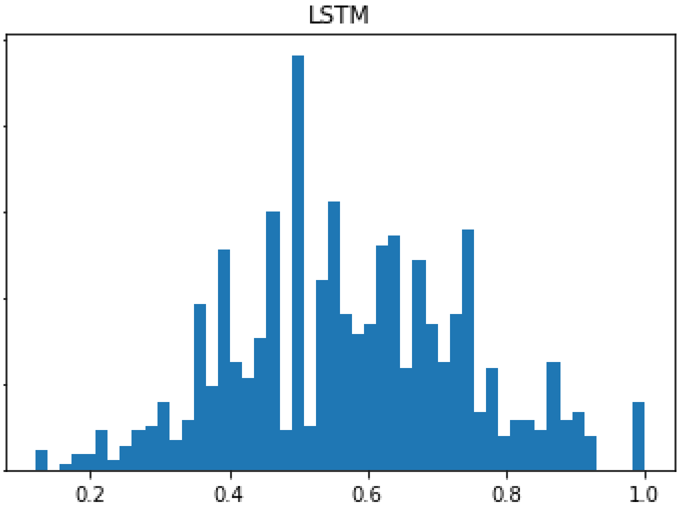


Figure 8. BLEU score distribution for IncceptionV3 with LSTM

Figures 9 depicts the distribution of BLEU scores over the entire dataset for the InceptionV3 BLSTM model. Again, more than 50% of the dataset is classified over a score of 0.5

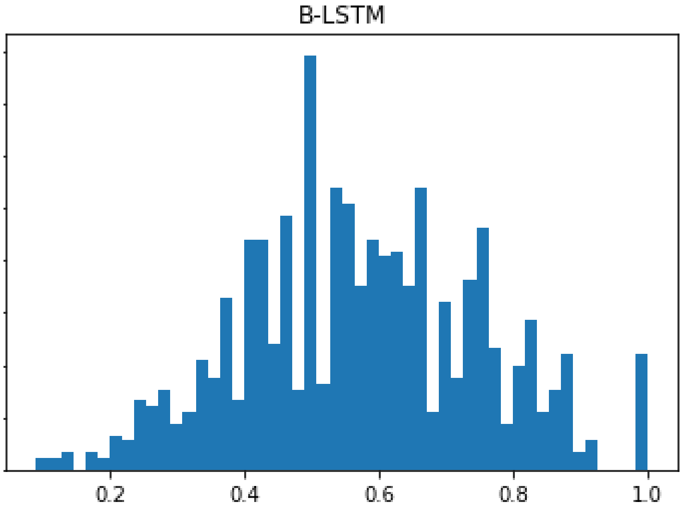


Figure 9.BLEU score distribution for IncceptionV3 with BLSTM

On the qualitative front, the MobileNet model was the least expensive in terms of computational resources. The InceptionV3 model with a BLSTM on the other hand, was the most expensive.

Conclusion:

Our initial hypothesis was that a BLSTM network would greatly outperform an LSTM network. Given that it is essentially a two time increase of the parameters in training the model, this seemed to be a valid assumption.

However, our results were not as obvious. While the BLSTM models did perform better, it was by a very small margin. As evident, an unexpected outcome is that the MobileNet model performed better than the InceptionV3 model. Despite being the smaller of the two and having a lower number of parameters to train, the MobileNet model with a BLSTM outperforms the InceptionV3 model with a BLSTM.

Based on our analysis, we observed both the models to have similar results for a BLEU evaluation. All scores ranged between 0.57 and 0.59. The MobileNet model is much more portable compared to InceptionV3 and has fewer parameters. Despite of being computationally inexpensive, it gave the best result for the Flickr8k dataset.

Thus, based on the results, the initial hypothesis isn’t as obvious. We believe the performance of these architectures are very subjective and would differ given varying use cases. While the BLSTM network does outperform an LSTM network, the miniscule gains do not justify the computational expense.

A possible approach to further identify the benefits of a BLSTM network, a larger dataset with a larger number of classes and a bigger word dictionary pool might give different results than obtained here.

A major limitation observed when working with these models is the computational requirement. Given enough resources, a much larger model can be trained against a much larger dataset. This would perhaps make the performance gains of a BLSTM over an LSTM far more profound and justifiable.

References:

1. https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/
2. http://www.aclweb.org/anthology/P02-1040.pdf
3. https://research.googleblog.com/2016/09/show-and-tell-image-captioning-open.html
4. https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Gatys_Image_Style_Transfer_CVPR_2016_paper.html
5. https://arxiv.org/abs/1604.00790
6. https://arxiv.org/abs/1704.04861
7. https://github.com/neural-nuts/image-caption-generator
8. https://commons.wikimedia.org/wiki/User:BiObserve (Raster version previously uploaded to Wikimedia) Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton (original)Eddie Antonio Santos (SVG version with TeX math) - https://commons.wikimedia.org/wiki/File:Long_Short_Term_Memory.png Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 6645–6649. IEEE, 2013., CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=59931189
9. https://github.com/neural-nuts/image-caption-generator
10. CS231n Winter 2016 Lecture 10 Recurrent Neural Networks, Image Captioning, https://youtu.be/cO0a0QYmFm8