INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

AUTOMATIC CAPTION GENERATION
CS 419 - INTRODUCTION TO MACHINE LEARNING

Aakash (180260001)
Mukta Wagle (18D070054)
Date: May, 2021

CONTENTS

1 Aim of the Project **2**

2 Dataset Description **2**

3 Pre-processing **3**

3.1 Image Data Preparation 3

3.2 Text Data Preparation 3

3.3 Training and Validation Data Preparation 4

4 Model Building **4**

4.1 Feature Extractor 4

4.2 Sequence Model 4

4.3 Merger 5

4.4 Attention Mechanism 5

4.5 Loss function 6

4.6 Optimizer 6

5 Evaluation **6**

6 Appendix **7**

1 AIM OF THE PROJECT

To generate word-by-word captions of images using CNN-based feature extraction with attention based Encoder-decoder LSTM caption construction.

2 DATASET DESCRIPTION

We have used the Flickr8k dataset for training, validation and testing the model. It contains a total of approximate 8,000 images, with at least 5 descriptions per image. The data is pre-divided into train, dev, and test sets. Flickr8k_text (2.2 MB) contains .txt files of image names mapped to their captions. Flickr8k_Dataset (1 GB) contains all the images.
Link to dataset :
Flickr8k_Dataset.zip
Flickr8k_text.zip

3 PRE-PROCESSING

3.1 Image Data Preparation

Image features are extracted using the VGG16 model that is pre-trained on ImageNet dataset (for classification). The image is encoded into a float vector through the output taken from the second last layer of VGG16. The dimensions of the encoded vector are (1,4096) and the datatype is numpy array.

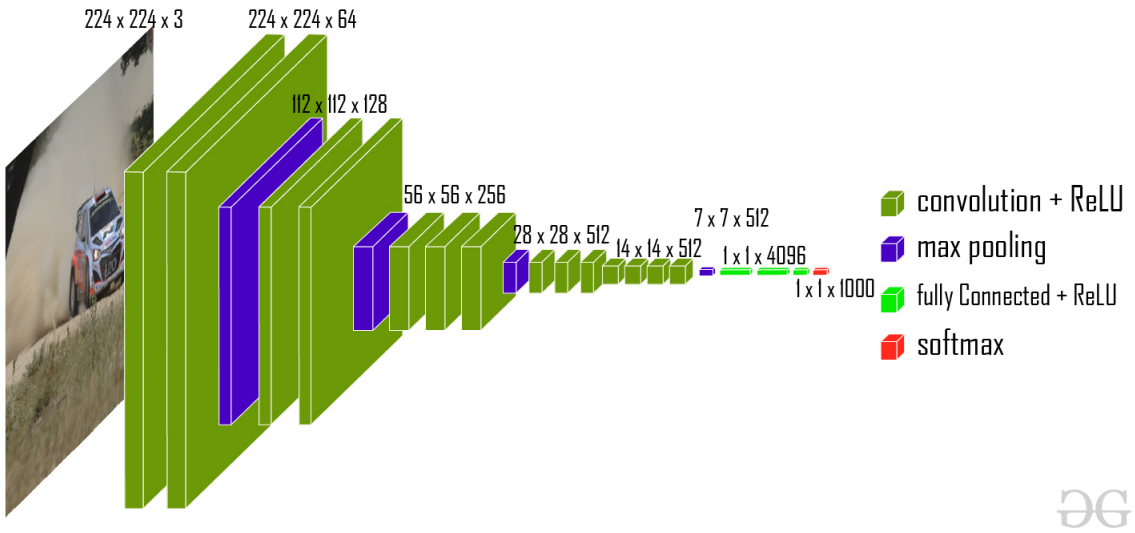


Figure 1: Source: VGG-16 CNN Model

3.2 Text Data Preparation

The .txt files containing image id and description are loaded, and a map (dictionary) is created from image id to list of descriptions corresponding to it. Then, we clean the text in the following ways in order to reduce the size of the vocabulary of words we will need to work with:

- Convert all words to lowercase.
- Remove all punctuation.
- Remove all words that are one character or less in length (e.g. 'a').
- Remove all words with numbers in them.

The clean_descriptions() function that, given the dictionary of image identifiers to descriptions, steps through each description and cleans the text. Next - Convert to a vocabulary of

words. Set (non-duplicate elements structure) of all words. Save all captions with image id (<id> *space* <caption>) line-by-line in the .txt file.

3.3 Training and Validation Data Preparation

The train and development dataset have been predefined in the Flickr_8k.trainImages.txt and Flickr_8k.devImages.txt files respectively, that both contain lists of photo file names. From these file names, we can extract the photo identifiers and use these identifiers to filter photos and descriptions for each set.
The strings ‘startseq‘ and ‘endseq‘ are added to the descriptions as identifiers for the start and end of the description.

We divide the data into 3 sets:

- Training set: 6000 images
- Validation set: 1000 images
- Test set: 1000 images

4 MODEL BUILDING

4.1 Feature Extractor

The Photo Feature Extractor model expects input photo features to be a vector of 4096 elements which are obtained from the pre-trained VGG16 model. These are processed by a Dense layer to produce a 256 element representation of the photo.

4.2 Sequence Model

The Sequence Processor model expects input sequences with a pre-defined length (34 words) which are fed into an Embedding layer that uses a mask to ignore padded values. This is followed by an LSTM layer with 256 memory units to process the input sequence and produce a 256 element output vector.

4.3 Merger

Both the input models produce a 256 element vector. Further, both input models use regularization in the form of 50% dropout. This is to reduce overfitting the training dataset, as this model configuration learns very fast.
The Decoder model merges the vectors from both input models using an addition operation. This is then fed to a Dense 256 neuron layer and then to a final output Dense layer that makes a softmax prediction over the entire output vocabulary for the next word in the sequence.

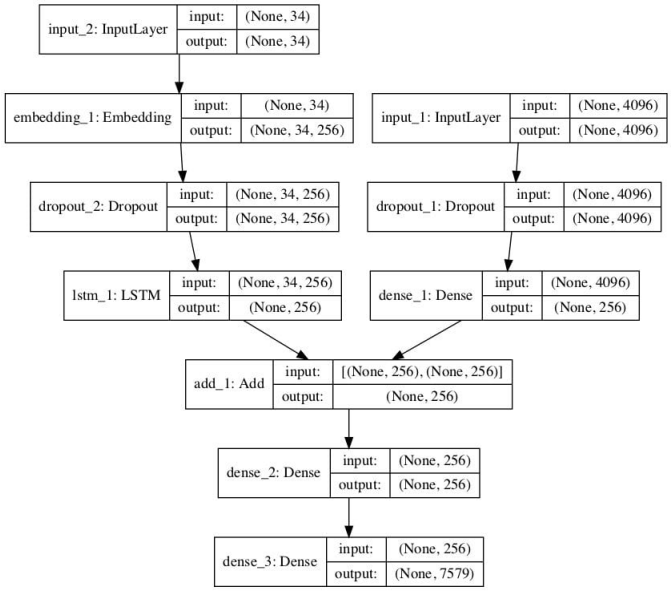


Figure 2: Caption Generation Model

4.4 Attention Mechanism

This is used in series with the LSTM in the sequential model. Instead of using the pre-defined Attention layer in keras, we build an attention mechanism using basic components as below:

1. Instead of using only the last layer of LSTM, we need to pay attention to the hidden layers as well. We can obtain the hidden states through the 'return_sequences=true' option in the LSTM layer.

2. The length of all word sequences is 34, which implies we have 34 time-steps in each cation generation instance. For each time-step, the output dimension is (None,256). A weighted sum of the 256 parameters is required to represent each word.

3. We can achieve this using a single-perceptron dense layer which will give a weighted

sum representing each of the 34 words. We flatten the output and pass it through softmax activation to make the 34 weights sum up to 1.

4. We then apply these attention-weights to the LSTM output with the help of the multiply layer, and wrap the multiply operation using lambda layer.

4.5 Loss function

Categorical cross-entropy loss for multi-class classification is used in the model.
Cross-entropy is a measure of the difference between two probability distributions for a given random variable or set of events. Categorical cross-entropy loss measures the difference between the real and predicted multi-class probability distributions.

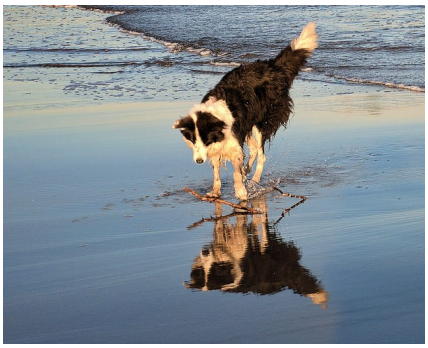
4.6 Optimizer

Adam (Adaptive Moment Estimation) is a variation of the gradient descent optimizer that computes adaptive learning rates for each parameter through standard back-propagation.

5 EVALUATION

The Caption Generation Model was evaluated on a dataset of 1000 images from the Flickr8k dataset with and without attention mechanism and the results are tabulated below.

BLEU Scores				
Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Attention	0.56	0.31	0.21	0.094
No Attention	0.53	0.28	0.18	0.077
Baseline	0.63	0.41	0.27	-



(a) black dog is running through the water

(b) man in red shirt is standing on the snow

(c) man in red helmet riding bike down dirt path

(d) man is climbing up rock

Figure 3: Examples of Generated Captions

6 APPENDIX

Link to the code.

References -

- Show, Attend and Tell: Neural Image Caption Generation with Visual Attention
- Craft your own Attention layer in 6 lines
- Image captioning with visual attention