ML2 FINAL PROJECT
GROUP 6

Professor Amir Jafari

Individual Final Report

Sharmin Kantharia

(December 7, 2020)

Introduction

This project aims to achieve Image Captioning using Deep Learning Techniques, mainly Convolutional Neural Networks and Long Short-Term Memory. The dataset used for this project is the Flickr8k dataset, found easily on Kaggle. Flickr8k, and others like it, such as the COCO dataset, have created the benchmark for sentence-based image captioning.

The scope of the project incorporates different domain under Artificial Intelligence such as Computer Vision, Deep Learning and Natural Language Processing. Applications and business value of this project includes:

- Image Search in Search Engines
- Image Segregation and Classification
- Text-to-Speech to aid visually impaired individuals
- Automatic Image annotation in Facial Recognition, E-commerce, etc.

The Flickr8 dataset has 2 main components – Images and their associated Captions. There are 8092 images, each having 5 captions in the results.csv file. Hence, there are a total of 40,460 captions. The dataset can be found here.

The project is divided into 5 main components:

1. Data Splitting – generating the train and test ids of the images
2. Text Preprocessing – preprocessing the captions before being fed into the LSTM model
3. Image Preprocessing – generating image features using pre-trained models
4. LSTM – Caption generator – generates captions
5. Model Evaluation – using various performance metrics

This individual report discusses the components Text Preprocessing and the ResNet50 model for Image Pre-processing.

Individual Work

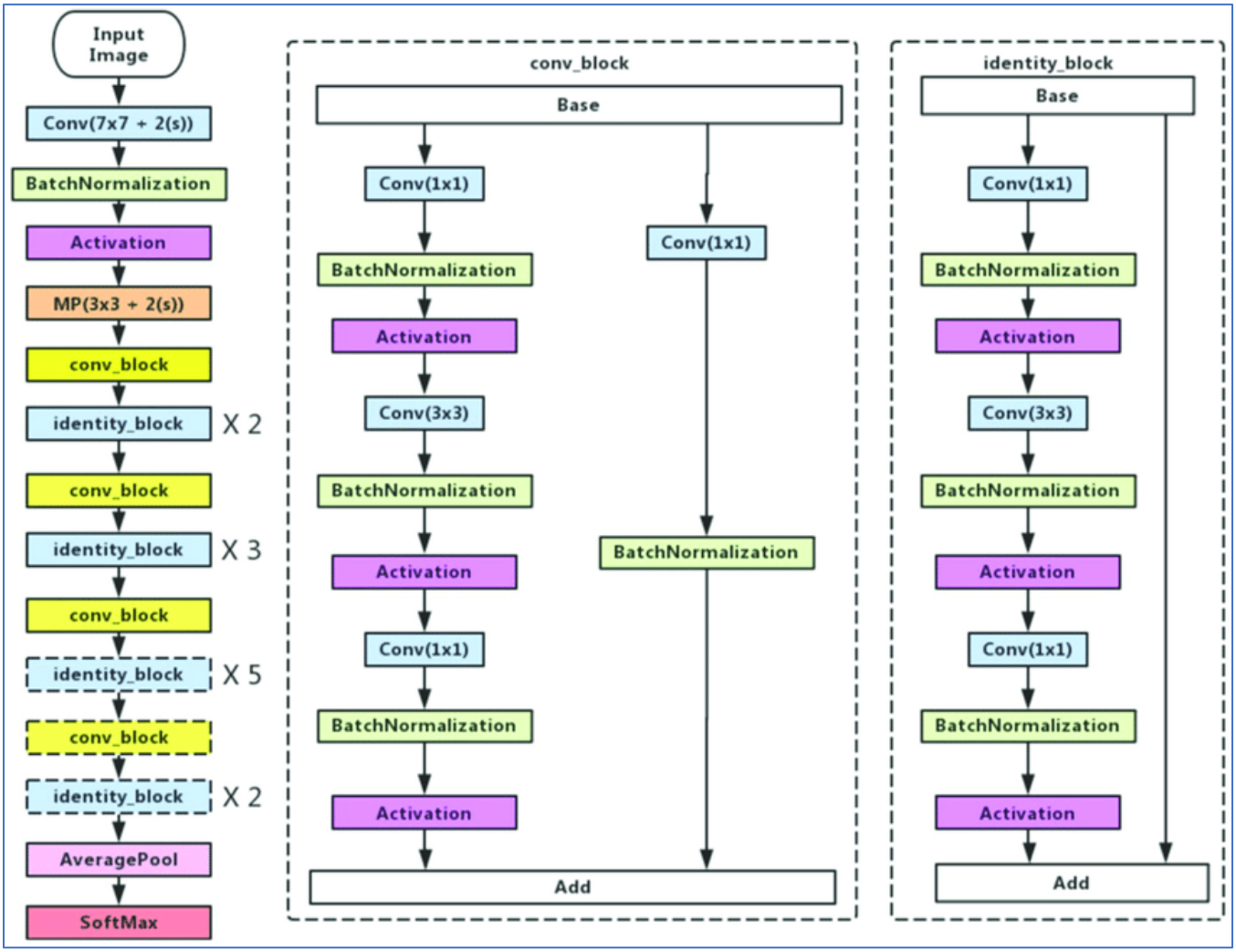
The LSTM model takes as input the image features and the word vectors generated during pre-processing. The text pre-processing component cleans the captions and extracts unique words against a threshold, which are then assigned to an integer. The maximum length of the caption

is also fed into the generator code for padding purposes. The words and their indices are then used to create the embedding matrix which provides the LSTM with sequenced captions.

A. Text Preprocessing
   1. load_doc() – this function opens, reads and closes the files being input.
   2. load_descriptions() – this function takes as input the captions of the entire dataset, tokenizes and splits the image id from the caption sentences using the "." separator. It creates a dictionary with the image id as key and the value is a list of corresponding captions. There are 8092 images.
   3. clean_descriptions() – this function takes as input the output generated in the previous step. It cleans the captions by removing punctuations, numbers and tokens with length < 1. It also converts all tokens to lower case. It outputs a dictionary of cleaned captions and a list of keys and values.
   4. to_vocabulary() – this function takes as input the dictionary of cleaned captions and provide a set of all the unique words in all the captions.
   5. save_descriptions() – saves the generated descriptions by writing to a file.
   6. load_set() – this function allows us to lad the pre-split image ids from text file.
   7. I then check if the split image ids are in the main list of images.
   8. load_clean_descriptions() – this function allows us to generate the train captions by taking as input the training image ids and the list of cleaned captions. This function also adds the 'startseq' and 'endseq' before and after each caption in the list. This is done to inform the LSTM model on when a caption begins and when it ends. The output is a dictionary of all cleaned, modified training captions.
   9. description_list() – this function takes as input the train captions and gives as out a list of all training captions.
   10. 'Vocab' contains all words that are greater than a threshold of 10, i.e., occurring more than 10 times.
   11. 'ixtoword' and 'wordtoix' are dictionaries created, which represents every unique word in the vocabulary with an integer and provides a list of all unique words in the train captions.
   12. to_lines() – this function creates a list of all the captions alone.
   13. max_length() – this function takes the training captions and returns the length of the longest caption. This is used further in generating the input sequences with padding.

B. Feature Extraction – ResNet50

The ResNet50 is a pre-trained model that was trained on the ImageNet dataset. For this project, I removed the last dense layer which was used for classification. The previous dense layer is used to extract the features. A final dense layer is used to get a vector of shape (2048,).



Results

The functions described above were used to clean captions and generate enumerated caption sequences which will be fed into the LSTM model for training. The captions were transformed as follows.

The image captions were loaded as follows and converted to a dictionary with image id as key and a list of all corresponding captions as values.

```
image,caption
1000268201_693b08cb0e.jpg,A child in a pink dress is climbing up a set of stairs in an entry way .
1000268201_693b08cb0e.jpg,A girl going into a wooden building .
1000268201_693b08cb0e.jpg,A little girl climbing into a wooden playhouse .
1000268201_693b08cb0e.jpg,A little girl climbing
```

```
'99679241_adc853a5c0': ['grey bird stands majestically on a beach while waves roll in .', 'large bird stands in the
water on the beach .', 'tall bird is standing on the sand beside the ocean .', "water bird standing at the ocean 's
edge .", 'white crane stands tall as it looks out upon the ocean .'], '997338199_7343367d7f': ['person stands near
```

The captions were then cleaned as follows.

```
 mountain scene', 'children getting ready to sled', 'people are sitting together in the snow'], '99679241_adc853a5c0':
 ['grey bird stands majestically on beach while waves roll in', 'large bird stands in the water on the beach', 'tall
 bird is standing on the sand beside the ocean', 'water bird standing at the ocean edge', 'white crane stands tall as it
  looks out upon the ocean'], '997338199_7343367d7f': ['person stands near golden walls', 'woman behind scrolled wall is
```

The unique words in the captions were identified as follows. There are 8680 unique words.

```
{'trips', 'kidsized', 'banner', 'counter', 'rabbits', 'creating', 'argues', 'cardigan', 'scooter',
Original Vocabulary Size: 8680
```

The image ids of the train set (split beforehand) are loaded. There are 6468 train images.

```
 '3265162450_5b4e3c5f1b', '2220175999_081aa9cce8', '3556598205_86c180769d', '2672354635_3a03f76486',
 '103195344_5d2dc613a3', '2695085448_a11833df95', '2278110011_ba846e7795', '3038941104_17ee91fc03',
 '2480668859_6f9b46be6a', '2704257993_d485058a5f', '242558556_12f4d1cabc', '2451988767_244bff98d1',
 '1394620454_bf708cc501'}
Dataset: 6468
```

The captions of the train images were cleaned and appended with a 'startseq' and an 'endseq' token as the start and end of each caption. This is done to provide a uniform starting partial caption to the language model.

```
patterns endseq', 'startseq writing on pad in room with gold decorated walls endseq'], '997722733_0cb5439472':
['startseq man in pink shirt climbs rock face endseq', 'startseq man is rock climbing high in the air endseq',
'startseq person in red shirt climbing up rock face covered in assist handles endseq', 'startseq rock climber in red
shirt endseq', 'startseq rock climber practices on rock climbing wall endseq']}
Length of all train captions : 32340
```

Input = Image_1 + 'startseq'; Output = 'the'

We then generate a list of all cleaned descriptions alone, which is used to find the length of the longest caption. This maximum length is taken into account while padding the partial caption sequence. The 'ixtoword' and 'wordtoix' dictionaries contain words and corresponding indices. The sequence of indices is passed to the model, rather than the actual text caption.

```
'tall bird is standing on the sand beside the ocean', 'water bird standing at the ocean edge', 'white crane stands tall
 as it looks out upon the ocean', 'person stands near golden walls', 'woman behind scrolled wall is writing', 'woman
standing near decorated wall writes', 'walls are covered in gold and patterns', 'writing on pad in room with gold
decorated walls', 'man in pink shirt climbs rock face', 'man is rock climbing high in the air', 'person in red shirt
climbing up rock face covered in assist handles', 'rock climber in red shirt', 'rock climber practices on rock climbing
 wall']
Description Length: 31
```

```
Preprocessed words 7838 -> 1725
Vocab size : 1726
Description Length: 31
```

Summary

The text preprocessing was done in order to generate numerical caption sequences which can be fed to the embedding layer in the final LSTM model. While there are different methods to clean the data, the functions above worked the best for the Flickr8k image captions. The pre-trained ResNet50 model was used to generate the image features with pre-determined weights. As explained in the final group report, did not perform as well as the VGG16 model, hence, it was not picked under the final selection.
Codes Used/Modified Percentage – 181 lines of code (found online), 2 lines of code (modified), 38 lines of code (added). Total = 81% (for text pre-processing and ResNet50 image pre-processing).

Conclusion

The text preprocessing done on the data was successful and proceeded to generate the accurate representations of the text captions. Further, NLTK packages and methods can be implemented to clean the text. While working with various pre-trained models, modifications can be made to further decrease the losses. Further work can involve using ResNet100, etc.

References

1. Learning CNN-LSTM Architectures for Image Caption Generation
2. Image Captioning
3. Show and Tell: A Neural Image Caption Generator
4. A Hierarchical Approach for Generating Descriptive Image Paragraphs
5. VizSeq: A Visual Analysis Toolkit for Text Generation Tasks
6. Recurrent Neural Network Regularization
7. Image Captioning - A Deep Learning Approach
8. https://hagan.okstate.edu/NNDesign.pdf
9. https://kharshit.github.io/blog/2019/01/11/image-captioning-using-encoder-decoder
10. https://towardsdatascience.com/image-captioning-in-deep-learning-9cd23fb4d8d2
11. https://colah.github.io/posts/2015-08-Understanding-LSTMs/