National University of Singapore
Department of Mathematics

DSA5204 Deep Learning and Applications
Semester 2, 2019/2020

Project Report

Author: Kee Wee Yang (A0099456L) (Group B)
Date: 08 May 2020

1. Introduction

Image captioning is an important aspect of computer vision and machine translation. In our project, we investigated an image captioning neural network presented in a research paper titled, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention" by Kelvin Xu et al (2016), and attempted to reproduce and experiment with the model created by the original authors. The findings are presented in the following sections of the report.

2. Image captioning neural network

An image captioning neural network is trained to determine objects in an image, identify the context of the image, and generate descriptive sentence in natural language. Attention mechanism is proposed by Kelvin Xu et al (2016) which allows for relevant image features to be focused on and translated to word as each word in a caption is generated in sequence.

The architecture of the image captioning neural network consists of an encoder and a decoder. The encoder consists of feature extraction of raw image by a convolutional neural network (CNN). The features extracted by CNN are passed to a recurrent neural network decoder which generates a caption. A schematic diagram of the encoder-decoder architecture is presented in the following figures.

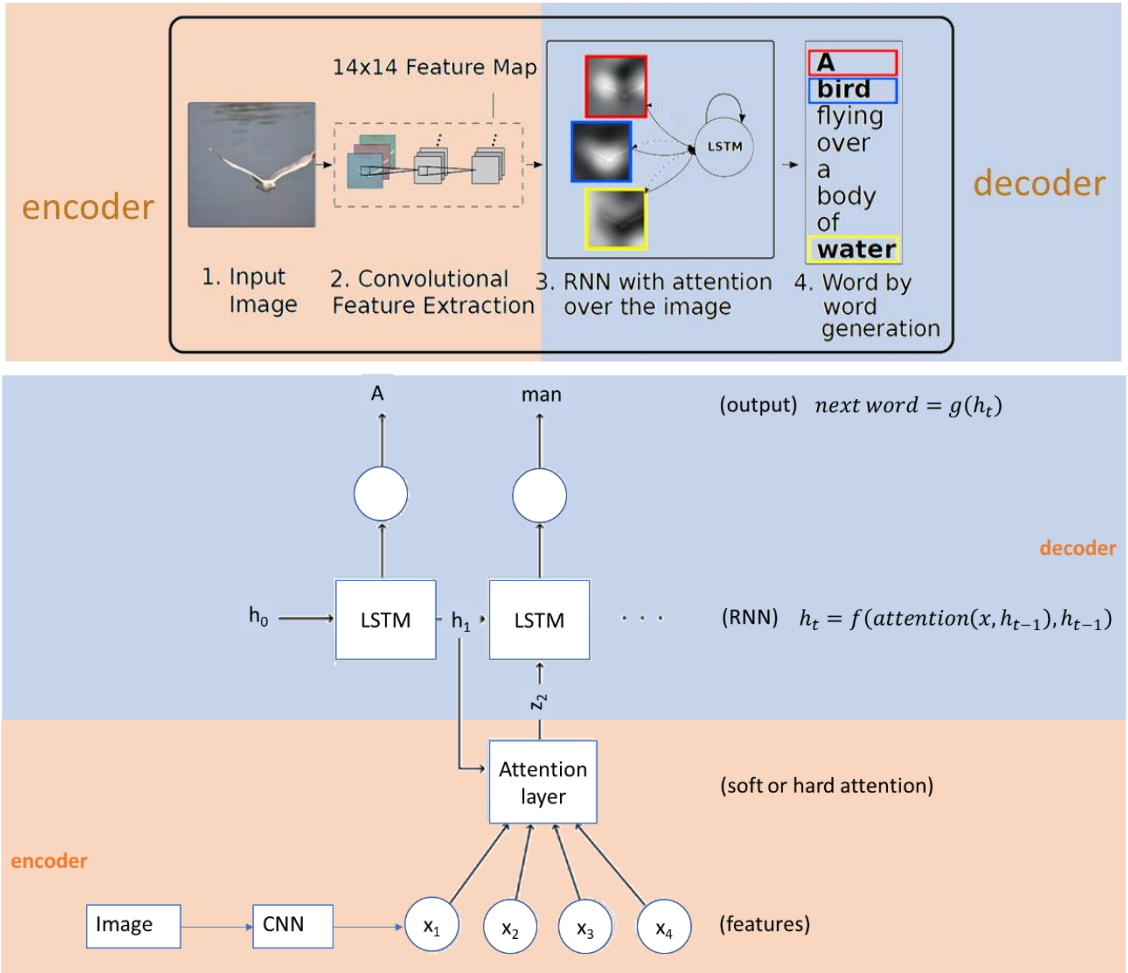

Figure 1 Overview of image captioning model (Above: Kelvin Xu et al, 2016; Below: Jonathan Hui, 2017)

The features extracted from CNN are passed through an attention layer. The outputs of the attention layer are fed to a Long-Short-Term Memory (LSTM) node. The LSTM node also takes input from the previous hidden state, and generates a current hidden state which is consequently used to generate the current word in the caption. A simplified mathematical representation can be expressed below (Jonathan Hui, 2017):

$$next\ word = g(h_t)$$
$$h_t = f(attention(x, h_{t-1}), h_{t-1})$$

where
h_t: hidden state to predict the current word
h_{t-1}: previous hidden state
x: image features

It is also noted that the attention layer takes input from the previous hidden state, which allows the attention mechanism to focus on the image features in a chronological manner, or in other words, allows the LSTM to generate the next word based on the context of the previous word. A pictorial representation of the attention mechanism is shown in the following figure, where white highlights correspond to areas of the image where attention is focused on.

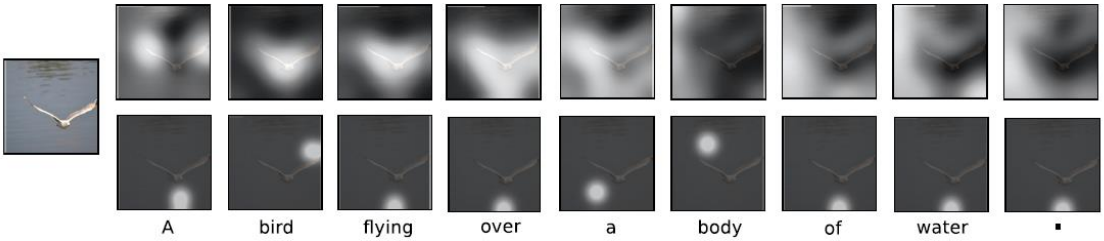

Figure 2 Attention mechanism (Above: soft attention; Below: hard attention) (Kelvin Xu et al, 2016)

A distinction is made between soft attention and hard attention mechanism. In soft attention, a weighted average of all image features is used as inputs to the decoder network, i.e., image features with higher weightage gets higher attention. Soft attention works in a deterministic way and uses standard back propagation to compute gradient descent. In hard attention, instead of a weighted average, only one hidden state is selected as input to the decoder network. Stochastic sampling is used as opposed to deterministic method in soft attention. To calculate the gradient descent correctly in the backpropagation, samplings are performed and results are averaged using the Monte Carlo method (Jonathan Hui, 2017).

3. Evaluation metric

Bilingual evaluation understudy (BLEU) score is adopted to evaluate the quality of captions generated by neural network (candidate caption), compared with reference captions written by human (reference caption). BLEU works by comparing match of words in the candidate caption and the reference caption. The match of words can be categorized by 'n-gram', where n is the number of words in a string. A unigram match is satisfied if any one word in the candidate caption is same as a word in the reference caption, and so on. For example,
Reference caption: A cat sits next to a dog.
Candidate caption: There is a cat and a dog.

n-gram	Sets	BLEU Score
Unigram (n=1)	"There", "is", "a", "cat", "and", "a", "dog"	4/7 = 57%
Bigram (n=2)	"There is", "is a", "a cat", "cat and", "and a", "a dog"	2/6 = 33%
Trigram (n=3)	"There is a", "is a cat", "a cat and", "cat and a", "and a dog"	0/5 = 0%
Four-gram (n=4)	"There is a cat", "is a cat and", "a cat and a", "cat and a dog"	0/5 = 0%

It can be seen from the illustration above that the lower the value of n, the higher likelihood of scoring a higher individual BLEU score. Weights can be assigned to each n-gram, and a weighted average BLEU score across the n-grams can be obtained. In both the paper and our reproduction, the following weightage are used.

Table 1 Weightage of each n-gram of BLEU scores

	Unigram	Bigram	Trigram	Four-gram
BLEU-1	1.0	0	0	0
BLEU-2	0.5	0.5	0	0
BLEU-3	0.3	0.3	0.3	0
BLEU-4	0.25	0.25	0.25	0.25

As image datasets often contain multiple reference captions per image, a corpus BLEU function, which allows comparisons with multiple reference captions, is used.

4. Reproduction and experimentation

The original authors produced results in Flcikr8k, Flickr30k, and COCO datasets. In our reproduction and experimentation, we first focus on Flickr8k dataset, which contains eight thousand images, each with five reference captions. We attempted to reproduce the results by using InceptionV3 convolutional network, and experiment with different optimizers and attention models. Subsequently, we used the same model trained on Flickr8k to train on COCO dataset. The comparisons with the original results are tabulated below.

Table 2 Comparisons of results

	Dataset	Convolutional network	Attention mechanism	Optimizer	BLEU-1 (%)	BLEU-2 (%)	BLEU-3 (%)	BLEU-4 (%)
Original	Flickr8k	VGGnet	Soft	RMSProp	67	44.8	29.9	19.5
			Hard	RMSProp	67	45.7	31.4	21.3
	COCO	VGGnet	Soft	Adam	70.7	49.2	34.4	24.3
			Hard	Adam	71.8	50.4	35.7	25
Reproduction	Flickr8k	InceptionV3	None	Adam	53.5	28.8	20	9.2
			None	RMSProp	50.1	23.2	14.9	5.8
			Soft (Luong, 2015)	Adam	53.6	28.4	19.1	8.5
			Soft (Bahdanau, 2015)	Adam	52.9	46.6	28	7.8
	COCO	InceptionV3	Soft (Bahdanau, 2015)	Adam	52.2	45.8	27.2	7.4

Firstly, two models are created without any attention mechanism. Adam and RMSProp optimizers are used in each model respectively. Adam optimizer yields better BLEU scores compared to RMSProp. Both models fall short in terms of BLEU scores compared to the original results, suggesting the importance of attention mechanisms in producing good captions. Subsequently, two models are created using two different versions of soft attention mechanisms, originated from Luong (2015) and Bahdanau (2015) respectively. It is noted that Bahdanau's (2015) soft attention mechanism is also used by the original authors. In our reproduction, both models use Adam optimizer. The latter model with Bahdanau's (2015) soft attention mechanism reproduced best BLEU scores overall. Compared with the original results, our best reproduction sees a lower BLEU-1 score, comparable BLEU-2 and BLEU-3 scores, and a lower BLEU-4 score. The differences could be due to the use of different convolutional network. It is noted that, in the original results, hard attention mechanism produced slightly better BLEU scores. However, we did not manage to reproduce results using hard attention mechanism and that should be further studied in the future.

5. Conclusion and further studies recommended

We have investigated an image captioning model presented by Kelvin Xu et al (2016) and attempted to reproduce and experiment with the model. With our best model, we obtained comparable but lower BLEU scores on Flickr8k and COCO datasets. The differences of the models lie in the use of different convolutional networks.

From our investigation, we can conclude that attention mechanism is important in producing good quality captions as seen from the comparison of BLEU scores between models with attention mechanism and models without attention mechanism.

For future studies, the followings are recommended:

- As reported by the original authors, hard attention mechanism produced slightly better BLEU scores on Flickr8k dataset. The reproduction of hard attention model could be further investigated. Furthermore, a different hard attention mechanism could be explored as further extension.
- In our reproduction, we have used InceptionV3 as compared to VGGnet used by the original authors, and differences in BLEU scores are observed. Therefore, the use of different convolutional networks could be further explored in attempt to improve BLEU scores.
- The original authors also employed another evaluation metric named METEOR. The reproduction of METEOR scores could be further investigated. Furthermore, the use of different evaluation metric could be explored as a further extension.
- The original authors reported that dropout and early stopping on BLEU score were the regularization strategies adopted in their models. Other regularization techniques could also be explored as a further extension in attempt to improve BLEU scores.

References

Bahdanau et al (2015). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473.*

Brownlee (2017). A Gentle Introduction to Calculating the BLEU Score for Text in Python. Retrieved from *https://machinelearningmastery.com/calculate-bleu-score-for-text-python/*.

Jonathan Hui (2017). Soft & hard attention. Retrieved from *https://jhui.github.io/2017/03/15/Soft-and-hard-attention/*.

Kelvin Xu et al (2016). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *International Conference on Machine Learning (pp. 2048-2057)*.

Lilian Weng (2018). Attention? Attention! Retrieved from *https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html#a-family-of-attention-mechanisms*.

Luong et al (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025.*

Seong (2019). Retrieved from *https://github.com/mybirth0407/Show-and-Tell-keras*.

TensorFlow Authors (2018). Retrieved from *https://github.com/tensorflow/docs/blob/master/site/en/tutorials/text/image_captioning.ipynb*.