Descriptive Image Captioning

Kenneth Chen

07/01/2019

Abstract

Convolutional neural network (CNN) revolutionizes the way images are detected in neural network. Recurrent neural network (RNN) proves to be more efficient at speech recognition and word embedding. There have been a number of successful attempts at translating an image into a descriptive text known as machine translation. A multimodal recurrent neural network (MRNN) combines the visual representation from CNN and sequential text generation from recurrent neural network (RNN). Here I implemented the MRNN on three benchmarks: Flickr8k, Flickr30k and MSCOCO2017 dataset. I demonstrate optimal layers of RNN and an embedding size for the corpus. Finally the crossover evaluation between three datasets gives an insight into how we can implement more advanced visual representations and develop a better metric to evaluate in the future.

1. Introduction

Image captioning by machine learning is an automatic process of generating texts based on the input image. Research in this area spans numerous domains, such as computer vision, natural language processing and machine learning. The benefits of automatic image captioning has many folds such as surveillance cameras generating texts based upon the scene or generated texts to visually impaired person and others. The interest in image captioning has resurfaced after significant progress in image classification such as AlexNet, VGG16 models and concurrent progress in natural language processing via deep learning model.

Given the image, the model predicts the caption describing the content of the image, eg, "a kid throwing a frisbee in an open field." Therefore the junction between a computer vision (CV) and natural language processing (NLP) creates a challenging testbed for multiple objects segmentation within an image and realistic interpretation of those objects in the image. In this paper, I implemented MRNN (Multimodal Recurrent Neural Network) to caption the given images. The MRNN model is implemented with two sub-networks: a deep CNN (VGG16) network for generating an image vector and a three-layered RNN network for a caption generation. Prior to that, image captioning was done with MRNN which consists of only one layer in RNN that receives the image vector. Most of the studies have done in Flickr and COCO dataset from 2014. The effectiveness of my model is validated on three benchmark datasets: Flickr8k, Flickr30k and MS COCO 2017 dataset (Hodosh et al., 2013; Young et al., 2014; Lin et al., 2014).

2. Related work

Image classification

During the span of past two decades, we have seen a significant improvement in image classification. In 1998, Yann LeCun and colleagues first laid out the convolutional neural networks for detecting a simple 10 digits classification (LeCun et al., 1998). An interest in image classification resurfaced after Alex and colleagues demonstrated a more powerful neural networks to classify images for as many as 1000 labels in ILSVRC-2012 competition (Krizhevsky et al., 2012). The model which is composed of many layers of convolution and pooling has rekindled a strong interest in deep convolutional neural networks for detecting images with more classes. The feat that researchers assumed in the beginning too broad and complicated even to classify an image with multiple objects in it. In 2014, Karen Simonyan and Andrew Zisserman developed a more refined image classification model, widely known as VGG16 and VGG19 (Oxford Visual Geometry Group), each comprised of 16 layers or 19 layers of convolution and pooling respectively (Karen Simonyan and Andrew Zisserman, 2014) (Figure 1). To this day, CNN has become the bread and butter of many successful image classification models.

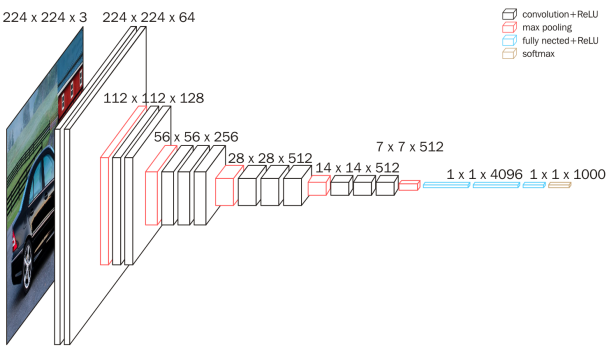

Figure 1. Deep convolutional neural networks, VGG16

Text classification and generation

Generally texts as simple as *good, bad, happy, angry* can be classified as either **positive** or **negative** sentiments. Emails can be classified as spams or hams by a simple model naive bayes (Sahami et al., 1998). Other models: SVM (Support Vector Machine), CNN and RNN are also found to be effective at classifying texts as the text description becomes longer and more subtle. In 2014, Kim showed that longer sentences can be classified by convolution over certain parts of the sentences (Kim, 2014). Among those models, RNN proves to be powerful at speech recognition and creating word embeddings. RNNs have been implemented in many machine translations to extract semantic information from the source and generate target sentences (Kalchbrenner and Blunsom, 2013).

3. Experiments

Dataset

Three benchmark datasets were used: Flickr8k [8091 images] (Hodosh et al., 2013), Flickr30k [31783 images] (Young et al., 2014) and MSCOCO 2017 (Lin et al., 2014). COCO has 118,287 training and 5,000 validation images. Each image from all dataset has 5 captions using Amazon Mechanical Turk (AMT). For COCO2017 dataset, the model was trained on COCOtrain. For Flickr8k and Flickr30k, 6,000 and 20,000 images were used for training respectively. I used 5000 and 10,000 images from Flickr8k and Flickr30k for validation and testing respectively.

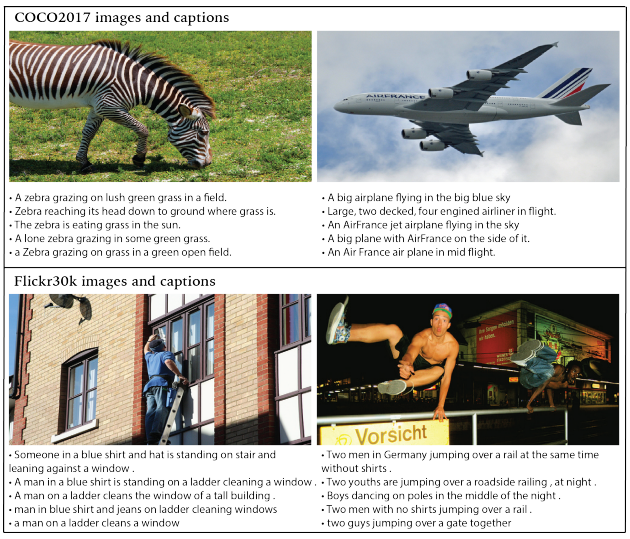

Figure 2. Samples of COCO and Flickr30k images

Multimodal Recurrent Neural Network

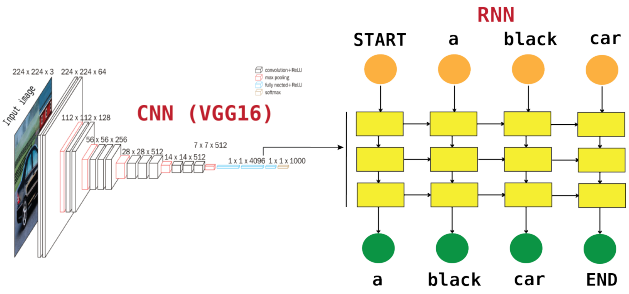

Figure 3. MRNN built upon CNN and RNN

To extract image features, VGG16 CNN was employed. VGG16 is made up of 16 convolutional layers and contains 138 million paramters. The architecture is uniform with 2 or 3 convolutional layers followed by maximum pooling layer. The final output is the probability distribution for classifiying 1000 labels (Simonya and Zisserman, 2014). The training on VGG16 using 4 GPU usually takes 2-3 weeks. Since the CNN weights are publicly available, I used VGG16 to extract Flickr and COCO image features. To retain the image features information, the vector output from penultimate layer in VGG16 model, known as fc7 (fully connected layer), was used as an image features, represented in 4096 vector. Those features are used as an initial state in recurrent layers (Figure 3).

In RNN sub-network, I implemented the skip-gram model (Mikolov et al., 2013). Skip-gram model has been implemented in image captioning before (Lazaridou et al., 2015). Here using the Flickr and COCO dataset with enriched description of images, this would allow the model to predict what kinds of surrounding words co-exist to describe an image. Here the model will predict the next word based on the current word. Basically, RNN sub-network has 6 layers: word embedding layer, 3 recurrent layers (GRU), dense layer and softmax output layer. A caption input is tokenized, lowercased and is in one-hot encoding format for 10000 corpus. At time t, the input vector w_i will have only one word encoded in 1 in respective vocabulary index while the rest are zero. The embedding layer will transform the input word vector, w_i, into relevant word representation from the embedding layer (10000 x 128 vector). The embedding layer encodes both the syntactic and semantic meaning of the words. In terms of word embeddings approach, most of the MRNN uses pre-computed word embedding vector such as Word2Vec (Karpathy et al., 2014; Frome et al., 2013; Kiros et al., 2014). However, I randomly initialized the word embedding layer and learn the word

vector representation during the training. Karpathy and Fei Fei in their note described that when they used the random initialization for word embedding, the model outperforms their previous implemention with Word2Vec, presumably due to inherent nature of word clustering in Word2Vec embedding.

The word vectors are then fed into GRU (Gated Recurrent Unit) (Figure 4). GRU has the input vector, x_t, and prior memory vector, h_{t-1}, reset gate, r_t and output vector, h_t. For h_{t-1} condition for the initial word, the image vector from CNN is used as an initial state. The approach is based on the concept that image features will prime the recurrent unit which word to generate at time t. The final output vector is softmaxed, which generates the tokenized word, the index of which is retrieved by `argmax`. The output word is used as a next input into the RNN sub-network to generate next word. This process continues until the output word is the tokenized end word *eeee*.

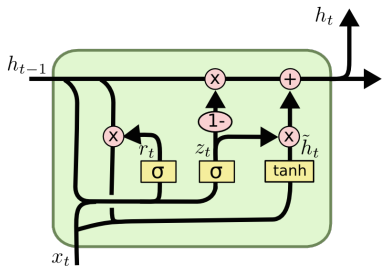

Figure 4.Gated Recurrent Unit

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$
$$r_t = \sigma(W_r \cdot [h_{t-1}])$$
$$\tilde{h}_t = \tanh(W \cdot [r_t \times h_{t-1}, x_t])$$
$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t$$

MRNN Training and Cost Function

In training Flickr and COCO images in captions generation, I used the log-likelihood as cost function since many studies used such cost function in sentence generation for efficiency and reliability. The log likelihood of the predicted word can be calculated as following:

$$\log_2 P(w_{1:L}|\mathrm{I}) = -\frac{1}{L}\sum_{n=1}^{L} \log_2 P(w_n|w_{1:n-1}, \mathrm{I})$$

where L is the length of the caption, $P(w_{1:L}|\mathrm{I})$ is the probability of the sentence $w_{1:L}$ given the image I. $P(w_n|w_{1:n-1}, \mathrm{I})$ denotes the probability of the current word given the previous word $w_{1:n-1}$ and an image I. Therefore, the cost function is the average log-likelihood of the words and a regularization parameter.

$$C = \frac{1}{N}\sum_{i=1}^{N_s} L_i \cdot \log_2 P(w_{1:L_i}^{(i)}|\mathrm{I}^{(i)}) + \lambda_\theta \cdot \|\theta\|_2^2$$

4. Results

Figure 5. Samples of predicted captions on COCO and non-COCO images. https://github.com/kckenneth/Image_captioning

The results from caption predictions on three datasets are shown in Figure 5. Basically, MRNN trained on COCO train dataset was validated on COCO validation dataset (5000 images), Flickr8k (8091 images) and Flickr30k (31783 images). Intuitively, the COCO trained model performs better on COCO validation set compared to Flickr datasets based on Bleu scores. The model trained on Flickr8k using 6000 images performs poorly among the three, presumably due to the smaller number of training images. During training, a number of parameters were tested, increasing the embedding size from 128 to 256 and 512, increasing the recurrent layers from 1 to 2, 3, 4 and up to 5. To avoid overfitting, dropout layer was implemented just before the final output layer in RNN sub-network. It was experimentally proven that embedding size of 128, and 3 layers of recurrent layers are optimal for COCO dataset. The training was achieved after 20 epochs to avoid overfitting on train dataset. However, the loss during training on Flickr dataset using either 2 or

MRNN trained on	COCO2017 [118287]				Flickr8k [6000]				Flickr30k [20000]			
test dataset	B-1	B-2	B-3	B-4	B-1	B-2	B-3	B-4	B-1	B-2	B-3	B-4
COCO[5000]	50.30	28.00	23.39	19.57	37.78	21.72	21.86	18.34	36.83	20.55	22.72	19.14
Flickr8k [8091]	40.40	23.20	21.83	18.39	44.78	23.41	21.44	17.89	45.54	23.65	22.17	18.80
Flickr30k [31783]	42.79	22.86	21.43	18.24	44.54	22.62	20.78	17.90	45.36	21.33	20.70	18.48

Table 1. Evaluation of model performance on three dataset. B-n: BLEU ngram score. More detailed scores for cumulative Bleu scores at scores.md in github.

MRNN trained on	COCO2017 [118287]						Flickr30k [20000]					
test dataset	R-1	R-2	R-3	R-4	R-L1	R-W	R-1	R-2	R-3	R-4	R-L1	R-W
COCO[5000]	30.88	8.25	2.67	1.04	34.44	15.74	27.37	5.19	1.12	0.37	31.22	13.92
Flickr8k [8091]	24.48	4.76	1.08	0.31	28.75	12.60	30.22	7.99	2.47	0.96	34.02	15.31
Flickr30k [31783]	23.64	5.01	1.29	0.42	27.77	11.74	27.34	6.95	2.14	0.87	30.91	12.96

Table 2. Rouge metric on two datasets. R-n: ROUGE ngram score, R-L: Longest common subsequence (LCS), R-W: weighted LCS.

3 recurrent layers never converge. Ultimately I only used one recurrent layer and trained for 60 epochs for Flickr8k and Flickr30k datasets.

Evaluation Metrics

Two most widely used metrics in machine translation were used here: BLEU and ROUGE scores. BLEU score is based on precision factor based on a quality of a predicted caption compared against a number of referenced sentences (Papineni et al., 2002). It has been reported that BLEU score up to 4 grams are correlated with human evaluation. Although BLEU score is easy to implement and execute, it also has many drawbacks such as penalizing an entirely different sentence if the predicted sentence is not found in a references although the prediction still makes sense to describe the image by human evaluation. However, due to time and labor constraint on evaluation, BLEU and ROUGE scores still generally serve as frontline evaluation metrics.

As shown in Figure 5 and Table 1, sentence predictions by COCO-trained model have more qualitative description compared to Flickr-trained model. For example, COCO-trained MRNN predicts a parrot image as *"a small bird on a tree branch with a blurry background"* whereas Flickr-trained model predicts *"a black dog is carrying a ball in its mouth."* By judging a number of captions on COCO and Flickr datasets, I observed a significant difference in quality of captions based on **concrete** and **abstract** level. For COCO dataset, majority of captions tend to describe images in a concrete way. For example, an image with a giraffe eating will have a caption "a giraffe eating leaves in the field". For Flickr dataset, an image with two women eating has a caption *Even though it's rather cool for outdoor dining , many people are enjoying the cafeteria-style food.* This kind of caption describes many level of abstract thought which are not easily found in dense image feature representations such as 4096 vectors. For example *even though* is a concrete description, rather a human sentiment on the next description. The *outdoor dining* is also abstract instead of a concrete description such as *a table.* Because of this nature of captions describing in each dataset, I found that COCO-trained MRNN tend to perform more qualitatively than Flickr-trained MRNN. This observation can be more evident when ROUGE score is employed, described in the following.

Novelty approach

I cross-examined the model performance on other test dataset. For example, the model trained on COCO2017 training dataset 118287 images was used to predict captions not only for COCO2017 validation dataset, but also for Flickr8k and Flickr30k dataset. The predicted caption was evaluated based on the reference captions (5 captions) available from each dataset. The underlying assumption is that a good model will predict sensibly for every image. I took such approach for all three dataset. I employed BLEU and ROUGE as evaluation metrics to check the performance of my models. Basically BLEU rougly measures the fraction of n-grams that are the same between a prediction and one or more references (Papineni et al., 2002). It also penalizes the short predictions by a brevity penalty term. Since evaluation up to 4-gram is still correlated with human evalution, I reported upto 4-gram. Cumulative BLEU scores are also reported in details here (https://github.com/kckenneth/Image_captioning/blob/master/scores.md). The other met-

ric, ROUGE, measures the recall statistics based on the number of words found in references and the predicted sentence (Lin, 2014). ROUGE-L identifies the Longest Common Subsequence (LCS) based statistics. The summary of all three models performance based on BLEU and ROUGE scores are shown in Table 1 and Table 2. The COCO-trained MRNN has 34.44 Rouge-L score. When the model was tested on Flickr8k and Flickr30k dataset, the recall score dramatically drops to 28.75 and 27.77 respectively, indicating that the prediction by COCO-trained model differs significantly from the references available for Flickr images.

5. Conclusion

Microsoft COCO dataset (Common Object in Context) provides a variety of images ranging from dogs sleeping on the porch to an aeroplane in the sky. Each image has human-labeled captions and they are more concrete description whereas Flickr captions are more abstract and describe human sentiment such as *enjoying, joking, playfully.* In this scenario, image detection by RPN (Region Proposal Network) would be much powerful at detecting several objects within an image and outperform better than classical CNN networks such as AlexNet, VGG and others.

6. References

Fang, H.; Gupta, S.; Iandola, F. N.; Srivastava, R. K.; Deng, L.; Dollár, P.; Gao, J.; He, X.; Mitchell, M.; Platt, J. C.; Zitnick, C. L. & Zweig, G. (2014), 'From Captions to Visual Concepts and Back.', CoRR abs/1411.4952 .

Frome, A.; Corrado, G. S.; Shlens, J.; Bengio, S.; Dean, J.; Mikolov, T. & others (2013), Devise: A deep visual-semantic embedding model, in 'Advances in Neural Information Processing Systems' , pp. 2121–2129 .

Hodosh, M.; Young, P. & Hockenmaier, J. (2013), 'Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics.', J. Artif. Intell. Res. 47 , 853-899.

Kalchbrenner, N. & Blunsom, P. (2013), Recurrent Continuous Translation Models., in 'EMNLP' , ACL, , pp. 1700-1709 .

Karpathy, A. & Li, F.-F. (2014), 'Deep Visual-Semantic Alignments for Generating Image Descriptions.', CoRR abs/1412.2306 .

Kim, Y. (2014), 'Convolutional neural networks for sentence classification', arXiv preprint arXiv:1408.5882 .

Kiros, R.; Salakhutdinov, R. & Zemel, R. S. (2014), Multimodal Neural Language Models., in 'ICML' , JMLR.org, , pp. 595-603 .

Krizhevsky, A.; Sutskever, I. & Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, in 'Advances in neural information processing systems' , pp. 1097–1105 .

Lazaridou, A.; Pham, N. T. & Baroni, M. (2015), 'Combining Language and Vision with a Multimodal Skip-gram Model.', CoRR abs/1501.02598 .

LeCun, Y.; Bottou, L.; Bengio, Y. & Haffner, P. (1998), Gradient-Based Learning Applied to Document Recognition, in 'Proceedings of the IEEE' , pp. 2278–2324 .

Lin, C.-Y. (2004), ROUGE: A Package for Automatic Evaluation of summaries, in 'Proc. ACL workshop on Text Summarization Branches Out' , pp. 10 .

Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C. L. & Dollár, P. (2014), 'Microsoft COCO: Common Objects in Context', CoRR abs/1405.0312.

Mikolov, T.; Chen, K.; Corrado, G. & Dean, J. (2013), 'Efficient estimation of word representations in vector space', arXiv preprint arXiv:1301.3781 .

Papineni, K.; Roukos, S.; Ward, T. & Zhu, W.-J. (2002), BLEU: a method for automatic evaluation of machine translation, in 'Proceedings of the 40th annual meeting on association for computational linguistics' , pp. 311–318.

Sahami, M.; Dumais, S.; Heckerman, D. & Horvitz, E. (1998), A Bayesian Approach to Filtering Junk E-Mail, in 'Learning for Text Categorization: Papers from the 1998 Workshop' , AAAI Technical Report WS-98-05, Madison, Wisconsin .

Simonyan, K. & Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', arXiv preprint arXiv:1409.1556 .

Socher, R.; Karpathy, A.; Le, Q. V.; Manning, C. D. & Ng, A. Y. (2014), 'Grounded Compositional Semantics for Finding and Describing Images with Sentences.', TACL 2 , 207-218.

Young, P.; Lai, A.; Hodosh, M. & Hockenmaier, J. (2014), 'From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions.' TACL