Long Short-Term Memory with Read-only Unit in Neural Image Caption Generator

Aghasi, Poghosyan
Institute for Informatics and Automation Problems of NAS RA
Yerevan, Armenia
e-mail: agasy18@gmail.com

Hakob, Sarukhanyan
Institute for Informatics and Automation Problems of NAS RA
Yerevan, Armenia
e-mail: hakop@ipia.sci.am

ABSTRACT

Automated caption generation for digital images is one of the fundamental problems in artificial intelligence. Most of the existing works use LSTM (Long Short-Term Memory) as a recurrent neural network cell to solve this task. After training, their deep neural models can generate an image caption. But there is an issue, the next predicted word of the caption depends mainly on the last predicted word, rather than the image content. In this paper, we present model that can automatically generate an image description and is based on a recurrent neural network with modified LSTM cell that has an additional gate responsible for image features. This modification results in generation of more accurate captions. We have trained and tested our model on MSCOCO image dataset by using only images and their captions.

Keywords

Deep learning, image caption generation, RNN, LSTM.

1. INTRODUCTION

Automated image caption generation is a difficult task that can help visually impaired people better understand the content of images on the web, also it can have a great impact on search engines and in robotics. This task is significantly harder than the well-studied image classification [1] or object recognition.

Generated image caption must contain not only image object names, but their properties, relations, and actions. Moreover, the generated caption must be expressed through natural language like English. This means, that already pre-trained neural language model needs an additional visual information to generate an image caption.

There are number of works approaching this problem. Some of them [2] [3] [4] offer to combine the existing image object detection and sentence generation systems. But there is a more efficient solution [5] that offers a joint model that takes an image and generates the caption, which describes the image adequately.

Last achievements in statistical machine translation were actively used in image caption generation tasks. The reason for this is mainly the proven achievement of greater results when using a powerful sequential model trained by maximizing the probability of the correct translation for the input sentence. These models [6] [7] [8] are based on Recurrent Neural Networks (RNNs). The model encodes variable length input into fixed length vector representation. This representation enables conversion of the input sentence into the target sentence or the input image into the target image caption. The last model was being trained to maximize $p(S|I)$ likelihood to generate the target sequence of words $S = \{S_1, S_2, \dots\}$ for an input image I, where each word S_t comes from a given dictionary, that describes the image adequately.

Today's public image datasets lack in detailed descriptions and variety of image scenes. These limitations cause RNN to predict subsequent word of the sentence by ignoring the image scene or objects. Thus, the next word prediction mainly depends on the previous word.

2. MODEL

Machine translation based models that can generate image descriptions actively use a recurrent neural network. We will maximize the probability of the correct caption for the given image.

$$\theta^* = \arg\max \sum_{(I,S)} \log p(S|I;\theta) \tag{1}$$

In formulation (1) θ represents the parameters of our model and S is its correct caption for the given image I. If we have a sentence $S = \{S_0, S_1, \dots S_N\}$ with the length of N, then we can apply the chain rule to calculate the joint probability (2) over $S_0, S_1, \dots S_N$,

$$\log p(S|I;\theta) = \sum_{t=0}^{N} \log p(S_t|I, S_0, \dots, S_{t-1};\theta) \tag{2}$$

where (I, S) is a training example pair. While training, we optimize the sum of the log probabilities for the whole training set using stochastic gradient descent [9].

$p(S_t|I, S0, \dots, S_{t-1};\theta)$ probability will correspond to the t step (iteration) of Recurrent Neural Network (RNN) based model. The variable number of words that are conditioned upon, up to $t - 1$ is expressed by a fixed length hidden state or memory h_t. After every iteration for the new input, x_t memory will be updated (3) by using a non-linear function f.

$$h_{t+1} = f(h_t, x_t) \tag{3}$$

For f we use a Long-Short Term Memory (LSTM), which has shown state-of-the-art performance on sequence generation

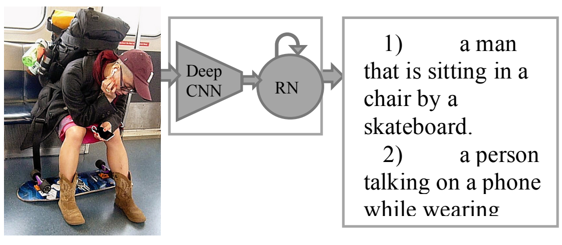

Figure 1. Model scheme that generates complete captions in natural language from an input image. (MSCOCO sample).

tasks, such as translation or image caption generation. Model (see Figure 1) consists of the feed forward deep convolutional neural network (CNN) that feeds RNN.
One of the best Convolutional Neural Networks (CNN) is *Google Inception* [10], that has been widely used in object classification and object detection tasks. Furthermore, there are works [11] that have done CNN transfer learning for object classification for such tasks as scene classification.
There are high-level features that describe image semantic content like objects, their properties and relations in CNN [12]. In this work, we will select *Mixed_7c* layer from *Google Inception* and append *average pooling* layer which will have 2048-dimensional output for image description. Also, we will append *fully connected* neural layer with N_e neurons, which will convert 2048-dimensional vector into N_e dimensional vector. N_e is a words embedding vector's dimensionality [13]. The output vector x_{-1} of fully connected layer will be the first feed vector for RNN.

2.1. LSTM

Long Short-Term Memory (LSTM) is an RNN cell. It helps in solving RNN training time problems like vanishing and exploding gradients [14], which is a significant problem for RNNs. LSTM is commonly used in machine translation, sequence generation and image description generation tasks. Work [5] uses recurrent neural network with an LSTM cell to generate image caption.

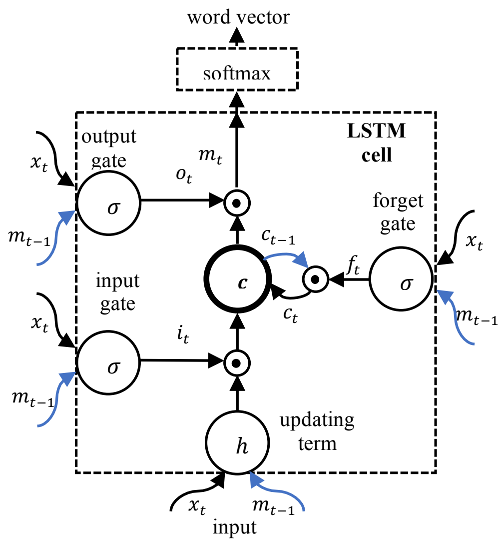

Figure 2. LSTM: the memory block contains a cell c which is controlled by three gates.

Constructionally LSTM is a memory cell c encoding knowledge at every iteration of what inputs have been seen up to this iteration. Later this knowledge is used for subsequent word generation (8, 9). Behavior of the cell is controlled by three *gates – input gate*, *output gate* and *forget gate*. Each gate is a vector of real number elements ranging from 0 to 1. In particular (see Figure 2), forget gate is responsible for controlling whether to forget the cell's old value, input gate controls the permission for reading a new input value and finally output gate controls the permission to output the new value from the cell. This is done by multiplying the given gate with the corresponding value (7, 8). The definition of the LSTM is as follows:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1}) \tag{4}$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \tag{5}$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1}) \tag{6}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}) \tag{7}$$

$$m_t = o_t \odot c_t \tag{8}$$

$$p_{t+1} = softmax(W_{pm} * m_t) \tag{9}$$

In (4-9) equations i_t, o_t, f_t are input, output and forget gates correspondingly, c_t is a cell memory in step t and m_t is an output of the LSTM for step (iteration) t. $W_{ix}, W_{im}, W_{fx}, W_{fm}, W_{ox}, W_{om}, W_{cx}, W_{cm}$ are trainable parameters (variables) of the LSTM. $\odot$ represents the product with a gate value. Sigmoid $\sigma(\cdot)$ and hyperbolic tangent $h(\cdot)$ are nonlinearities of the LSTM. The equation (9) will produce a probability distribution p_{t+1} over all words in the dictionary, where W_{pm} is a trainable parameter that represents words embedding.

2.1. Training

The LSTM model is trained to predict the probability for the next word of an image caption after it has observed all the previous words in the captions and image features. For easier training LSTM is represented in unrolled form (see Figure 3), which is a copy of the LSTM memory for the image and each word of the sentence. Also all LSTMs share the same parameters.

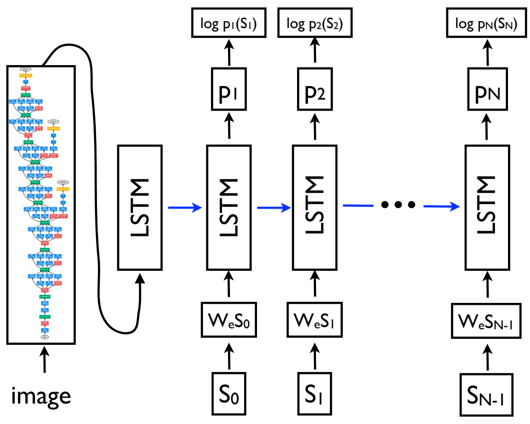

Figure 3. LSTM model combined with a CNN image embedder (as defined in [5]) and words embedding.

Thus, x_{-1} is the first input for the first LSTM. Initial state of the LSTM is c_{-1} zero-filled memory. For the next LSTMs, inputs correspond to the word embedded vectors. Also, all recurrent connections are converted into feed-forward connections.
For the input image I and the image's true caption $S = \{S_0, S_1, \dots S_N\}$, the unrolling procedure is:

$$x_{-1} = CNN(I) \tag{10}$$

$$x_t = W_e S_t \tag{11}$$

$$p_{t+1} = LSTM(x_t) \tag{12}$$

where each word S_t is the row of square identity $N_d{\times}N_d$ matrix at corresponding index, where N_d is the dictionary size. Also, S_0 is a special start word and S_N is a special stop word which indicates the start and the end of the sentence.
Note that both the image and the words are mapped to the same space. Vision CNN's last fully connected layer maps the image content to the embedding space. Also words are embedded by words embedding W_e, where W_e is a trainable parameter with $N_e{\times}N_d$ dimensionality.
Loss [5] is the sum of the negative log likelihood of the correct word at each step:

$$L(I,S) = -\sum_{t=1}^{N} \log p(S_t) \tag{13}$$

After training the model by minimizing loss with gradient descent, we will have all the parameters of the LSTM, the top layer of the image embedder CNN and word embedding W_e.

2.3. LSTM with Read-only Unit

After one million training iterations on MSCOCO [15] image dataset, we got a loss function graphic (see green graphic in Figure 4).

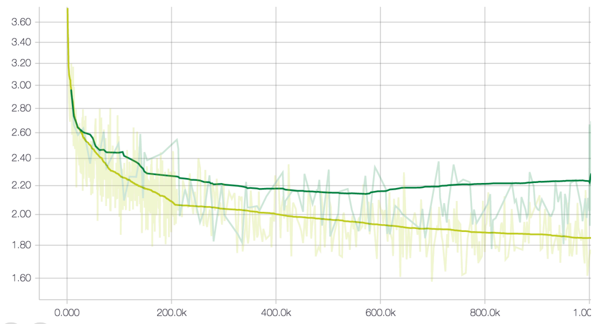

Figure 4. Train loss function, LSTM – green, LSTM with read-only memory – yellow

Experiments have shown that the LSTM's next prediction mainly depends on the previous word, that is why the LSTM generates words that are not associated with an input image. We will add an additional unit to the LSTM, that will help to create a new LSTM state that depends on the image content. This additional unit will be a new gate the value of which will be appended in the state calculation (see Figure 5).

$$r_t = \sigma(W_{rx}x_t + W_{rm}m_{t-1}) \tag{14}$$

$$c_t = f_t \odot c_{t-1} + r_t \odot x_{-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}) \tag{15}$$

In (14) r_t is the read-only gate with W_{rx} and W_{rm} additional trainable parameters. New state c_t is calculated as shown in (15). After retraining with the new gate, we can see the new training loss (see yellow graphic in Figure 4).

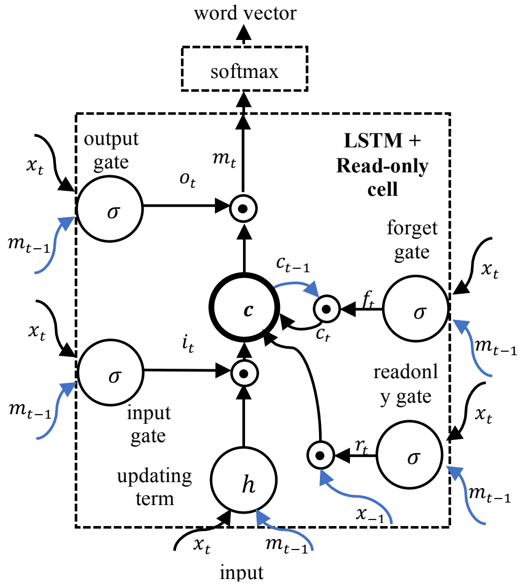

Figure 5. LSTM: the memory block contains a cell c which is controlled by four gates(additional read-only memory).

We inference by using BeemSearch and already trained parameters to generate an image caption as presented in work [5]. Some examples of inference are presented bellow (see Figure 6).

LSTM

a) a couple of bikes parked next to each other.
b) a couple of bikes that are next to a building.
c) a couple of bikes parked next to a building

LSTM + Read-only cell

a) a building with a red door and a white door.
b) a building with a red door and a white door
c) a building with a red brick wall and a white building

LSTM

a) a woman in a bikini standing on a beach.
b) a woman in a bikini standing on a beach holding an umbrella.
c) a young girl in a bikini standing on a beach.

LSTM + Read-only cell

a) a woman in a dress and a hat on a beach.
b) a girl in a dress and a hat on a beach
c) a woman in a dress and a hat on a beach

LSTM

a) a woman sitting at a table with a laptop.
b) a woman sitting at a table with a laptop
c) *a woman sitting at a table in front of a laptop .*

LSTM + Read-only cell

a) a woman sitting at a table in a dress
b) a woman is sitting at a table with a hat.
c) *a woman sitting at a table with a hat on .*

LSTM

a) a group of people flying kites in the sky.
b) a group of people flying kites in the air.
c) *a group of kites flying in the sky.*

LSTM + Read-only cell

a) a bunch of animals that are in the grass.
b) a bunch of animals that are standing in the grass.
c) *a bunch of animals that are standing in the sand .*

Figure 6. Image caption generated by LSTM and LSTM with Read-only Unit models.

4. CONCLUSION

In this work a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) cell and a Read-only Unit has been developed. An additional unit has been added to work [5] that increases the model accuracy. Two models, one with the LSTM and the other with the LSTM and Read-only Unit have been trained on the same MSCOCO image train dataset. The best (the loss is minimum) middle loss values are 2.15 for LSTM and 1.85 for LSTM with Read-only Unit. MSCOCO image test dataset has been used for testing. Loss values for LSTM and LSTM with Read-only Unit model test are 2.05 and 1.90, accordingly. These metrics have shown that the new RNN model can generate the image caption more accurately.

REFERENCES

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," arXiv:1409.0575, 2014.

[2] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier and D. Forsyth, "Every picture tells a story: Generating sentences from images," in *ECCV*, 2010.

[3] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg and T. L. Berg, "Baby talk: Understanding and generating simple image descriptions," *CVPR,* 2011.

[4] A. Karpathy and L. Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions," *IEEE,* vol. 39, no. 4, pp. 664 - 676, 2015.

[5] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and Tell: Lessons learned from the 2015 MSCOCO Image Captioning Challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PP, no. 99, July 2016.

[6] D. Bahdanau, K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv:1409.0473, 2014.

[7] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine transla- tion.," *EMNLP,* 2014.

[8] I. Sutskever, O. Vinyals and Q. Le, "Sequencetosequence learning with neural networks.," *NIPS,* 2014.

[9] L. Bottou, "Large-scale machine learning with stochastic gradient descent.," in *Physica-Verlag HD*, 2010.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the inception architecture for computer vision," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 2818--2826, 2016.

[11] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng and T. D. Decaf, "A deep convolutional activation feature for generic visual recognition," in *ICML*, 2014.

[12] A. Poghosyan and H. Sarukhanyan, "Image Visual Similarity Based on High Level Features of Convolutional Neural Networks," *Mathematical Problems of Computer Science,* vol. 45, pp. 138--142, 2016.

[13] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *Proceedings of Workshop at ICLR*, 2013.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation,* vol. 9, no. 8, 1997.

[15] T. Lin, M. Maire, S. Belongie, J. Hays and P. Perona, "Microsoft coco: Common objects in context.," in *European Conference on Computer Vision*, 2014.