CS698O: Project Report

Abhishek Verma(14026), Ayush Tulsyan(14167)
abhivm@iitk.ac.in, ayusht@iitk.ac.in

Modified Stacked Attention Networks for VQA using Visual grounding of Phrases

Code available here

November 16, 2017

Abstract

The model we have used for visual question answering is called stacked attention networks (SANs) that learn to answer natural language questions from images. SANs use semantic representation of a question as query to search for the regions in an image that are related to the answer.We have implemented a multiple-layer SAN in which we query an image multiple times to infer the answer progressively. We also tried different models, namely LSTMs, LSTMs with attention, Seq2Seq LSTM for converting query phrase to a vector.

Keywords – SAN, stacked, attention, LSTM, Seq2Seq, CNN, RNN

1 Introduction

Computer vision is the science and technology of obtaining models, meanings, and control information from visual data. This field has influenced the field of Artificial intelligence greatly. The two important fields of computer vision are computational vision and machine vision amongst which computational vision deals with recording, analyzing and trying to understand a visual perception.

One of the important problem is the field of computer vision is visual question answering(VQA). VQA is the area which tries to find answer to a problem given an image and a question using a combination of natural language processing and computer vision techniques. One of the major steps in solving the VQA problem is visual grounding of phrases and features because to identify objects in the image, the subject is interacting with, one has to identify/visually ground the subject first along with the set of objects. Visual grounding problem takes image-sentence pairs as input and learns to visually ground (i.e., localize) arbitrary linguistic phrases. One may think that this problem is similar to the object detection problem. But that's not the case. In object detection, given a set of labels, we need to locate them in the visual whereas in visual grounding problem, given a natural language phrase, we need to analyze it and localize its context in the visual. In this project, well deal with images.

Visual grounding has a direct use in many fields such as surveillance AIs (to check if a certain activity has taken place in a video), daily household AIs(to find the location about which AI has been given instructions), etc.

2 Related Work

Image question answering is closely related to image captioning [1], [2], [3], [4], [5], [6], [7]. In [4], the system first extracted a high level image feature vector from GoogleNet and then fed it into a LSTM to generate captions. The method proposed in [2] went one step further to use an attention mechanism in the caption generation process. The approach proposed in [3] first used a CNN to detect words given the images, then used a maximum entropy language model to generate a list of caption candidates, and finally used a deep multimodal similarity model (DMSM) to rerank the candidates. Instead of using a RNN or a LSTM, the DMSM uses a CNN to model the semantics of captions.

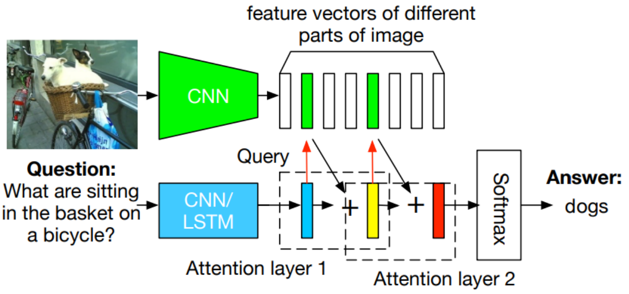
Several data sets have been constructed in [8], [9], [10], [11] either through automatic generation based on image caption data or by human labeling of questions and answers given images. Among them, the image QA data set in [9] is generated based on the COCO caption data set.

Several image QA models were proposed in the literature. [12] used semantic parsers and image segmentation methods to predict answers based on images and questions. [8] and [10] both used encoder-decoder framework to generate answers given images and questions. They first used a LSTM to encode the images and questions and then used another LSTM to decode the answers. [11] first encoded questions with LSTMs and then combined question vectors with image vectors by element wise multiplication. [13] used a

CNN for question modeling and used convolution operations to combine question vectors and image feature vectors[14].

3 Stacked Attention Networks (SANs)

The overall architecture of SAN is shown in *Figure 1*[14]. We will be describing all the components of our model in this section.


feature vectors of different parts of image
CNN
Query
Question:
What are sitting in the basket on a bicycle?
CNN/ LSTM
Softmax
Answer:
dogs
Attention layer 1
Attention layer 2

(a) Stacked Attention Network for Image QA

Figure 1: **SAN** - Stacked Attention Network

3.1 Image Model

This is the same model as mentioned in [14]. The image model uses a CNN to get the representation of images. Specifically, the VGGNet (VGG19) is used to extract the image feature map from raw image. We choose the features from the last pooling layer, which retains spatial information of the original images. We first rescale the images to be 448 x 448 pixels, and then take the features from the last pooling layer, thus getting a dimension of 512x14x14, as shown in *Figure 2*[14]. 14 x 14 is the number of regions in the image and 512 is the dimension of the feature vector for each region.

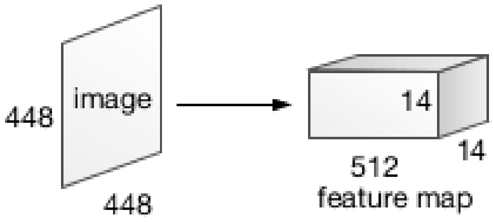


Figure 2: **Image Model** - based on CNN (VGG19)

We then use a single layer perceptron to transform each feature vector to a new vector that has the same dimension as the question vector.

3.2 Question Model

We use LSTMs to transform query phrase to a vector. Given the question in form of one hot vector representation of words at their corresponding positions, we first embed the words to a vector space through an embedding matrix. Then for every time step, we feed the embedding vector of words in the question to LSTM as shown in *Figure 3*[14]. Different models of LSTMs that we have used in our project are:

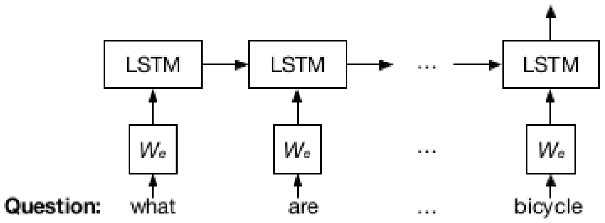


Figure 3: **Question Model** - using LSTM

3.2.1 Simple LSTM

They are mentioned in [19].

3.2.2 Attention based LSTM

These LSTMs were presented in [20]. Its mechanism can be understood through *Figure 4*.

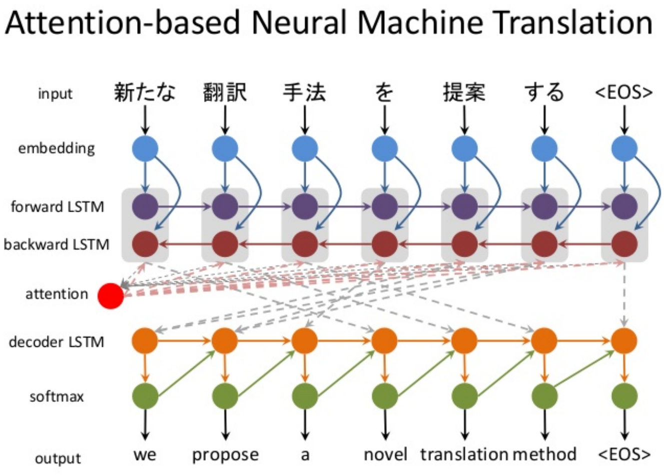


Figure 4: **LSTM** - attention based

3.3 Stacked Attention Networks

Given the image feature matrix and the question feature vector, SAN predicts the answer via multi-step reasoning. In many cases, an answer only related to a small region of an image. Therefore, using the one global image feature vector to predict the answer could lead to sub-optimal results due to the noises introduced from regions that are irrelevant to the potential answer. Instead, reasoning via multiple attention layers progressively, the SAN are able to gradually filter out noises and pinpoint the regions that are highly relevant to the answer. Mathematical details can be found in original paper.[14]

4 Experiments

4.1 Data set

We evaluated our model on MS-COCO version 2 dataset which can be found on visualqa website. COCO-QA is based on the Microsoft COCO data set, the authors first parse the caption of the image with an off-the-shelf parser, then replace the key components in the caption with question words for form questions. There are 78736 training samples and 38948 test samples in the data set. These questions are based on 8000 and 4000 images respectively. There are four types of questions including Object, Number, Color, and Location. Each type takes 70%, 7%, 17% and 6% of the whole data set, respectively. All answers in this data set are single word.

4.2 Results

Following results were obtained when LSTMs without attention were used:

Methods	**Accuracy**	**WUPS0.9**	**WUPS0.0**
Guess	6.9	17.2	72.8
SAN(1,2)	36.4	45.3	77.1
SAN(2,2)	43.6	48.3	84.1

With attention mechanism incorporated in LSTM, the results improved significantly:

Methods	**Accuracy**	**WUPS0.9**	**WUPS0.0**
Guess	6.9	17.2	72.8
SAN(2,1)	47.4	45.4	85.2
SAN(2,2)	53.7	53.8	87.4

Here, *SAN(x,y)* is a notation for using *x* SAN layers and *y* attention layers in LSTM.

4.3 Observations

- Accuracy increased when attention mechanism was added in LSTM. [*Figure 5*]
- Training error decreased and accuracy increased with an increase in number of LSTM layers till a point after which it started increasing.

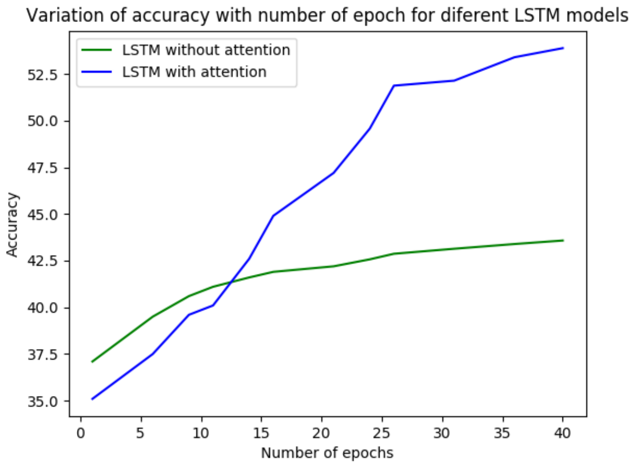


Figure 5: **Accuracy** - for different LSTM models

- Training error decreased when number of attention layers were increased from one to two. However it started increasing as the number of attention layers were increased further from two. The case with accuracy is vice versa. [*Figure 6*]

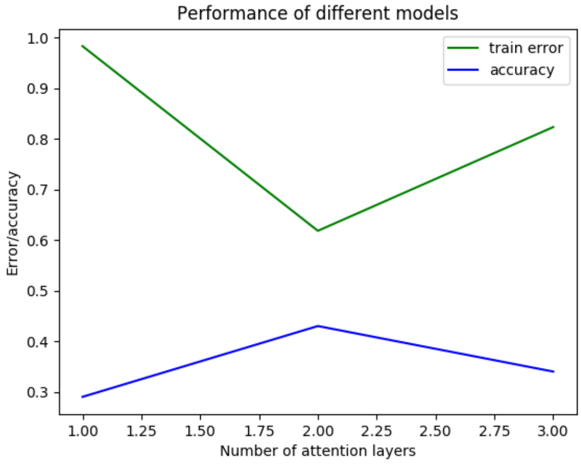


Figure 6: **Performance v/s number of attention layers** - Effect of number of SAN attention layers

- Training error started increasing after 25th epoch when we kept the learning rate constant. We analyzed it and started diminishing the learning rate at as the epochs proceeded. This resulted in an almost constant training error after 30th iteration. [*Figure 7*]
- Variation of error with number of epochs when diminishing learning rate mechanism was encapsulated. [*Figure 8*]
- Results of different optimization methods. [*Figure 9*]

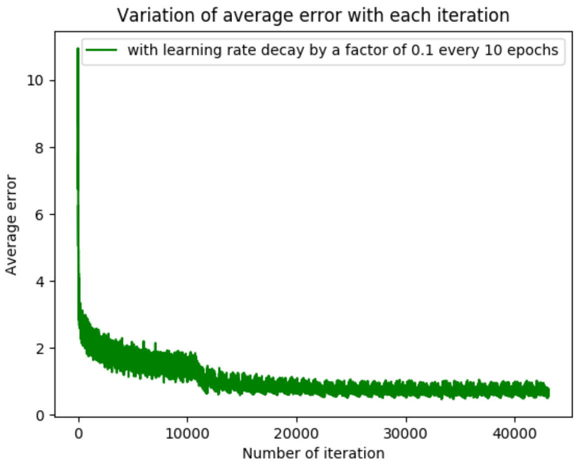


Figure 7: **Training error** - variation with each epoch

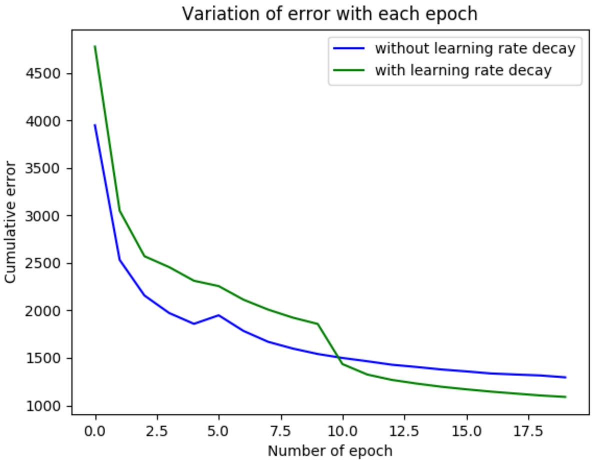


Figure 8: **Error variation** - Effect of diminishing learning rate

5 Conclusion

We implemented stacked attention networks for visual question answering in PyTorch(this was previously unavailable). SAN uses a multiple-layer attention mechanism that queries an image multiple times to locate the relevant visual region and to infer the answer progressively. The Stacked attention mechanism serves as a tool for visual grounding of question phrases.

References

[1] X. Chen and C. L. Zitnick. *Learning a recurrent visual representation for image caption generation*. arXiv preprint arXiv:1411.5654, 2014.

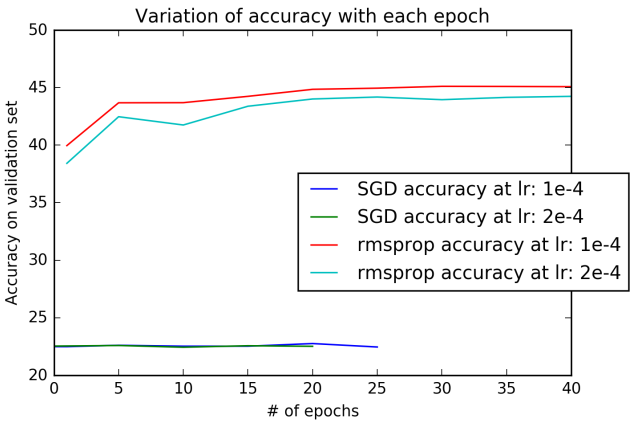


Figure 9: **Optimization methods** - Effect of optimizer on accuracy

[2] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. *Show, attend and tell: Neural image caption generation with visual attention*. arXiv preprint arXiv:1502.03044, 2015.

[3] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. Platt, et al. *From captions to visual concepts and back*. arXiv preprint arXiv:1411.4952, 2014.

[4] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. *Show and tell: A neural image caption generator*. arXiv preprint arXiv:1411.4555, 2014.

[5] R. Kiros, R. Salakhutdinov, and R. S. Zemel. *Unifying visual-semantic embeddings with multimodal neural language models*. arXiv preprint arXiv:1411.2539, 2014.

[6] A. Karpathy and L. Fei-Fei. *Deep visual-semantic alignments for generating image descriptions*. arXiv preprint arXiv:1412.2306, 2014.

[7] J. Mao, W. Xu, Y. Yang, J. Wang, and A. Yuille. *Deep captioning with multimodal recurrent neural networks (m-rnn)*. arXiv preprint arXiv:1412.6632, 2014.

[8] M. Malinowski, M. Rohrbach, and M. Fritz. *Ask your neurons: A neural-based approach to answering questions about images*. arXiv preprint arXiv:1505.01121, 2015.

[9] M. Ren, R. Kiros, and R. Zemel. *Exploring models and data for image question answering*. arXiv preprint arXiv:1505.02074, 2015.

[10] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. *Are you talking to a machine? dataset and methods for multilingual image question answering*. arXiv preprint arXiv:1505.05612, 2015.

[11] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. *Vqa: Visual question answering*. arXiv preprint arXiv:1505.00468, 2015.

[12] M. Malinowski and M. Fritz. *A multi-world approach to question answering about real-world scenes based on uncertain input*. In Advances in Neural Information Processing Systems, pages 1682âĂŞ1690, 2014.

[13] L. Ma, Z. Lu, and H. Li. *Learning to answer questions from image using convolutional neural network*. arXiv preprint arXiv:1506.00333, 2015

[14] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng and Alex Smola. *Stacked Attention Networks for Image Question Answering*. arXiv:1511.02274

[15] Fanyi Xiao, Leonid Sigal and Yong Jae Lee. *Weakly-supervised Visual Grounding of Phrases with Linguistic Structures*. arXiv:1705.01371

[16] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. *Learning deep features for discriminative localization*. In CVPR, 2016.

[17] S. Hochreiter and J. Schmidhuber. *Long short-term memory*. Neural computation, 1997.

[18] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. *Dropout: a simple way to prevent neural networks from overfitting*. Journal of Machine Learning Research, 2014.

[19] Sepp Hochreiter and JÃijrgen Schmidhuber. *Long Short-Term Memory*. Journal of Neural Computation, 1997.

[20] Minh-Thang Luong, Hieu Pham and Christopher D. Manning. *Effective Approaches to Attention-based Neural Machine Translation*.