MODIFYING STACKED ATTENTION NETWORKS ARCHITECTURE FOR VQA

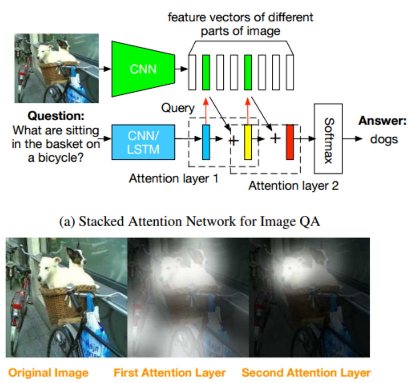
Final Report, Group 10, 08 Nov 2016
Prakhar (13485), Preetansh (13508), Viswanadh (13561)
CS698N: Recent Advances in Computer Vision, Jul-Nov 2016
Instructor: Gaurav Sharma, CSE, IIT Kanpur, India

1 INTRODUCTION AND PROBLEM DESCRIPTION

Visual Questioning and Answering is one of the most challenging problems of Computer Vision. An Image QA system takes and input image and a question pertaining to the image and produces an answer as the output. An AI system capable of answering such questions finds its applications in image-database searching, searching within an image (eg: a non expert querying a medical image in some remote areas), surveillance, assisting blind people, and for making a more advanced chat-bot as well, which may serve as customer-care service provider.

2 RELATED LITERATURE

The SOTA paper we chose for our problem area is, *Stacked Attention Networks(SANs) for Image Question Answering*[1] which proposes multiple-step reasoning for the problem of Image QA. SAN consists of stacked attention layers, which reads the image as 14x14 regions and gives an attention probability to each region. The overall architecture of the SAN is shown in following figure, and consists of three main components: the *Image* model, the *Question* model and the *Stacked attention* model.



A CNN, VGGNet is used by the image model to extract the image feature map f_I from a raw image I, VGGNet is used. Question model extracts query vector v_Q from the Question. SAN layers utilize both image features and query vector to obtain probability vector p_M which then modifies the query vector. This is done recursively K times by K attention layers, and then the final query vector u^K is used to final classification. Note that SAN treats VQA as a classification task, hence our training/testing set consists of one-word-answer questions only.
Many papers are using Attention mechanisms on images for improving the performance. Out of current top performers on VQA Open-Ended task, four out of five use some kind of attention mechanism. [2] uses a co-attention mechanism to incorporate attention-probabilities on both image feature matrix well as query feature vector. Inspired by them, we try to develop our own co-attention mechanisms over SAN.

3 DATASET AND CODE USED

3.1 VQA dataset

The VQA dataset [10] is created through human labeling. The data set uses images in the MS COCO image caption data set.

The actual idea in the paper is to send images into VGGNet model and the output of last maxpooling layer is saved as the feature to the corresponding image. VGG-Net 19 takes $224 \times 224 \times 3$ and outputs $7 \times 7 \times 512$ at the last maxpooling layer but the authors mentioned that they are sending $448 \times 448 \times 3$ and extracted the output at the last maxpooling layer which has dimensions $14 \times 14 \times 512$ at the last maxpooling layer.

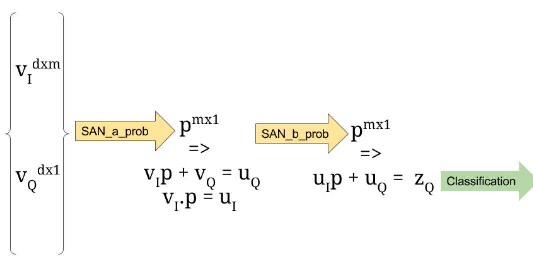
3.2 Existing codes and Libraries used

We have used author's implementation of SAN model as our base for further implementations. The Dataset used is VQA. Authors have also globally uploaded the image-features extracted from VGG-NET, for the VQA dataset. Those are also directly used. Rest of the code for all the modifications over it for implementing our mechanisms is completely written by us.

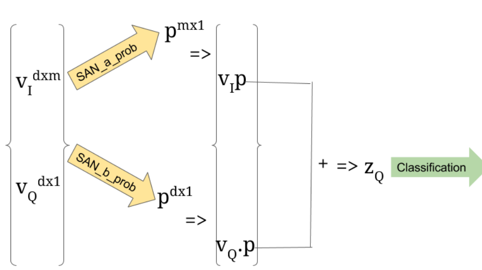
4 Our Modifications of SAN

Four different architectures have been implemented by us over SAN. All of them take image feature v_I and question feature v_Q to finally obtain a modified query vector z_Q that is sent for the classification task ('.' in figure implies element-wise multiplication, and circular $\oplus$ implies row-wise sum).

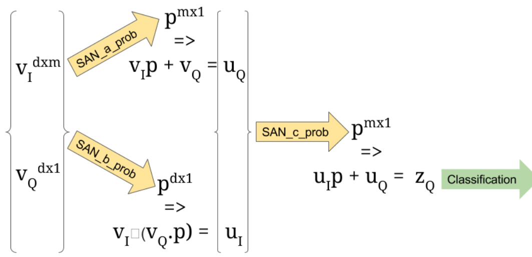
- **Model 1** : SAN_a obtains an m dimensional probability vector p^m to modify query vector v_Q to u_Q as given in original paper, and then i^th region of v_I is multiplied by i^th element of p^m to obtain modified image matrix u_I. SAN_b is normal SAN layer on u_I and u_Q to get z_Q



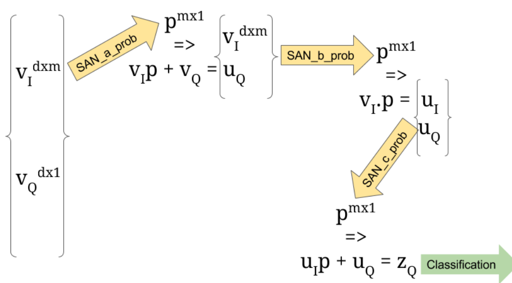
- **Model 2(Parallel Attention)** : SAN_a obtains p^m to simply obtain modified query v_Ip. SAN_b obtains probability vector p^d that modifies query to get v_Qp. Both these are added to get z_Q



- **Model 3** : SAN_a obtains SAN_a obtains p^m to simply obtain modified query u_Q. SAN_b obtains p^d to modify image matrix v_I by adding $p.q$ to each row, to get u_I. Finally, u_I and u_Q are passed through normal SAN layer to get z_Q.



- **Model 4(Alternate Attention)** : It is just an alternate version of Model 2 where SAN_a is used to get u_Q, then u_Q and v_I are passed in SAN_b to finally get u_I, and then we have normal SAN_c over u_I and u_Q to get z_Q



5 Results

- SAN (Actual Model Accuracy) = 0.525(after 50 epochs)
- Model 1 = 0.522 (after 50 epochs)
- Model 2 = 0.355 (after 20 epochs)
- Model 3 = 0.476 (after 4.12 epochs)
- Model 4 = 0.0 (just started)

6 Discussions

We explored multiple different attention mechanisms to enhance the initial SAN model. Our focus during the project was primarily on enhancing the attention mechanism and we did not focus much upon the tweaking with different feature representations for the images and question. Analyzing the results available as we write this report we found that the attention mechanism is useful only till a certain extent and advancements solely in attention beyond a certain accuracy are not feasible. Thus, work must also be done on the enhancements of feature representations to improve performance in VQA by improving the language and visual models so that suitable attention mechanism can applied.

7 Comparison Section

Project Proposal	Work done till Mid Semester	New work for Final Evaluation
Till Mid-sem evaluation: Reproduce results of the SOA paper on our system. Give a demo during mid-sem presentation. Try to implement the paper in *Tensor Flow*. **For Final project:** We proposed to try some other attention mechanisms to enhance the model such as a a co-attention network that develops attention on both question and image, to improve the performance. As the model currently gives only one word answers, since it solves a classification problem. So we'll try to replace the classification softmax in the end to give multiple word answers as well.	We were able to run a full-scale version of their model on a GPU machine. Also, we were able to qualitatively analyze the Stacked Attention Network Layers by visualizing the probabilities using the results. We found that the code using TensorFlow was taking around 4 times the time taken by Theano implementations and thus we aborted the TensorFlow implementation idea and we continued with the original implementation in *Theano*.	We developed *four additional models* after thinking about possible shortcomings in the original model suggested in [1]. All these models tried to use the attention mechanism in a different way and for this we explored how others were using attention mechanism [2]. We were not able to work on the idea of extending the model to predict multi-word answers as the dataset in itself did not have sufficient multi-word answer samples. If we try using any LSTM weights which is pre-trained with large text corpus would lead to prediction of abrupt answers independent of the image. So we focused on improving by experimenting with different mechanisms

References

[1] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Smola, *Stacked Attention Networks for Image Question Answering*. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June, 2016.

[2] Jiasen Lu, Jianwei Yang, Dhruv Batra, Devi Parikh, *Hierarchical Question-Image Co-Attention for Visual Question Answering*. CoRR, abs/1606.00061, 2016.

[3] Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, Richard Socher, *Ask Me Anything: Dynamic Memory Networks for Natural Language Processing*. CoRR, abs/1506.07285, 2015.

[4] Ilija Ilievski, Shuicheng Yan, Jiashi Feng, *A Focused Dynamic Attention Model for Visual Question Answering*, CoRR, abs/1604.01485, 2016.

[5] Abhishek Das, Harsh Agrawal, C. Lawrence Zitnick, Devi Parikh, Dhruv Batra, *Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions?*. CoRR, abs/1606.03556, 2016.

[6] Martín Abadi, et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. CoRR, abs/1603.04467, 2016.

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

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

[9] M. Malinowski and M. Fritz. *A multi-world approach to question answering about real-world scenes based on uncer- tain input.* In Advances in Neural Information Processing Systems, pages 16821690, 2014

[10] 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

[11] Simonyan, Karen, and Andrew Zisserman. *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556 (2014).

[12] github/ksimonyan *19-layer model from the arXiv paper: "Very Deep Convolutional Networks for Large-Scale Image Recognition"* https://gist.github.com/ksimonyan/3785162f95cd2d5fee77.

[13] github/JamesChuanggg *Torch Implementation for Stacked Attention Networks for Image Question Answering* https://github.com/JamesChuanggg/san-torch.

[14] github/zcyang *Source code for Stacked attention networks for image question answering.* https://github.com/zcyang/imageqa-san.