Visual Question Answering Using Various Methods

Shuhui Qu
Civil and Environmental Engineering Department
Stanford University
California, CA, 94305
shuhuiq@stanford.edu

Abstract

This project tries to apply deep learning tools to enable computer answering question by looking at images. In this project, the visual question answering dataset[1] is introduced. This dataset consists of 204,721 real images, 614,164 question and 50,000 abstract scenes, 150,000 questions. Various methods are reproduced. The analysis on different models are presented.

1 Introduction

Due to the rapid development of deep learning methods for visual recognition, natural language processing, computers could perform various complex and difficult tasks. One of the most important tasks is to have computer combining various tools for high-level scene interpretation, such as image captioning and visual question answering. With the emergence of large image dataset, text, questions, visual question answering by computers has been made possible. In general, the visual question answer system are required to answer all kinds of questions people might ask relate or not related with the image. Building a system that could properly answer questions would be important to the development of artificial intelligence.

Various methods have been proposed to deal with the problem. Due to the time limit, in this project, I manage to reproduce, analyze and compare previous works on this problem. These methods includes: pure yes, LSTM question only, CNN + LSTM, DMN.

The accuracy of thess models are evaluated by the vqa dataset, which contains 204,721 real images, 614,164 question and 50,000 abstract scenes, 150,000 questions.

Section 2 discusses the related works on image captioning and question answering methods. Section 3 introduces various module for question answering tools. The testing and result discussions are in sections 4. Then the report concludes with section 5.

2 Related Work

As the development of deep learning, there are a lot of studies related with high-level scene interpresations. Karpathy and Feifei[3] developed a model that generate natural language descriptions of images and regions that could learn about the inter-modal correspondances between language and visual data. They align the convolutional neural network with bidirectional recurrent neural network. Johnson et al.[2] presented convolutional localization network for dense captioning that could process an image with a single efficient forward pass. Recently, after the work by Agrawal et al[1], that they introduce the task of free-form and open ended visual question answering, there's emergence of works on tackling these tasks. Agrawal et al. also provided some initial effort on these problems. Zhou et al.[6] developed Bowing as a simple baseline for this task. Kumar et al.[4] presented dynamic memory network tools for natural language processing. Xiong et al.[5] further developed

dynamic memory network to enable visual and textual question answering. In this project, I am trying to reproduce most of these works.

3 Methodology

In order to successfully address the task, there are four main modules: image feature extraction, question understanding, answer generation and feature filters. The first three modules are essential for the project, which help to understand images and question and reasoning possible answers to the correct result. Depend are various techniques, the forth module are applied to improve the final accuracies.These possibe techniques includes normalization, BOW, episodic memory network and etc. Due to time limit, I only apply the episodic memory network in this project.

3.1 Visual Feature Extraction

In this project, a pretrained convolutional neural network based model VGGNet is applied to extract image features. Depend on different question answering model's architecture, the feature layer is

Figure 1: Image feature extraction

selected accordingly. For instance, the fully connected layer "fc3" is selected for CNN+LSTM model as shown in the red box of figure 1. This layer has 4096 parameters and could be input to answer generation module directly. For DMN model, the "conv5_3" layer is selected as shown in blue box of figure 1. This layer has $14 \times 14 \times 512$ parameters. During implementation for extracting "conv5_3" layer, various problems are encountered: out of memory, out of disk space, cannot write such large matrix. I also tried to extract features during runtime as well. However, this results unreasonable long training time.

3.2 Question Understanding

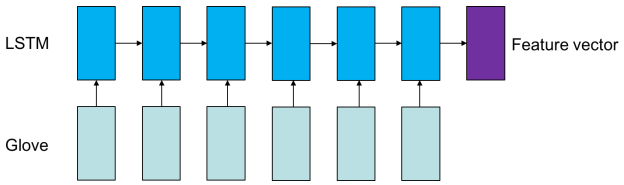

Figure 2: Question feature extraction

The module uses LSTM to extract question features q (the final hidden state of LSTM). Pretrained Glove is applied to represent each word.

3.3 Answer Generation

The answer generation module receives both question feature(or filter question feature) and image feature(or filtered image feature). These two features are then concatenated and input to an LSTM module to a sequence of words. Cross entropy loss on answers is applied to train the network.

3.4 Episodic Memory Network

There are various filters for images and questions. In this project, the episodic memory network[5] is applied that functions as the attention mechanism on images input. No filters on questions feature is applied, e.g. iBOW, etc.

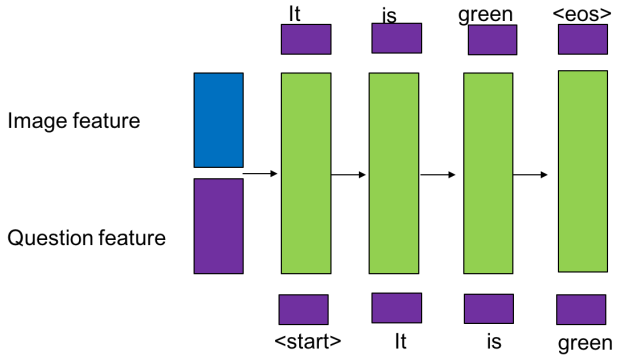

Figure 3: Answer generation

3.4.1 Input Module for EMN

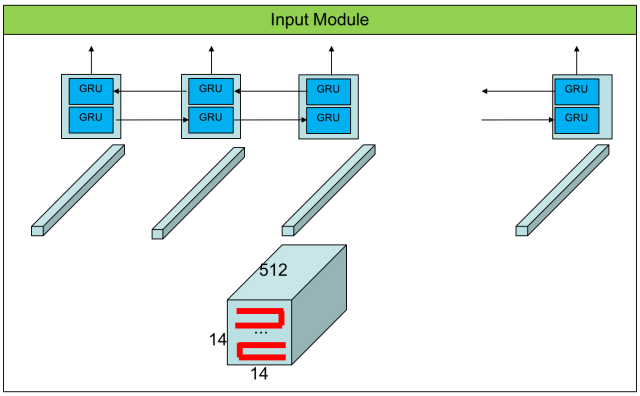

Figure 4: image input module

In this input module, there are two components instead of three parts described in [5] visual feature extraction and input fusion layer. The visual feature extraction is the the same as the previou section 3.1. I directly use numpy.reshape to concatenate the 196 local regional vectorsf_i instead of snake shape. After local feature extraction, they are then input to a bi-directional GRU system to produce globally aware input facts F. By using bi-directional GRU, information are propagated through all neighbors.The GRU function is as follow:

$$\begin{aligned} u_i &= \sigma(W^{(u)}X_i + U^{(u)}h_{i-1} + b^{(u)}) \\ r_i &= \sigma(W^{(r)}X_i + U^{(r)}h_{i-1} + b^{(r)}) \\ \tilde{h_i} &= tanh(Wx_i + r_i \circ Uh_{i-1} + b^{(h)}) \\ h_i &= u_i \circ \tilde{h_i} + (1 - u_i) \circ h_{i-1} \end{aligned} \tag{1}$$

where σ is sigmoid activation function. Then, for each local feature, forward hidden state and backward hidden state are combined as the output of the input module.

$$\begin{aligned} \overrightarrow{h_i} &= GRU_{fwd}(f_i, \overrightarrow{h_{i-1}}) \\ \overleftarrow{h_i} &= GRU_{bwd}(f_i, \overleftarrow{h_{i-1}}) \\ F_i &= \overrightarrow{h_i} + \overleftarrow{h_i} \end{aligned} \tag{2}$$

3.4.2 EMN Mechanism

The output of the input module $F = [F_1, F_2, ..., F_i, ..., F_N]$, where $N = 196$ in this case, are the input to the episodic memory module to provide further focusing attention information related with

question feature q. This is to find the relation between question feature and input F. The relation is represented by the attention gate g_i^t.

$$\begin{aligned} z_i^t &= [F_i \circ q; F_i \circ m^{t-1}; |F_i - q|; |F_i - m^{t-1}|] \\ Z_i^t &= W^{(2)} tanh(W^{(1)} z_i^t + b^{(1)}) + b^{(2)} \\ g_i^{(t)} &= \frac{exp(Z_i^t)}{\sum_{k=1}^{M_i} exp(Z_k^t)} \end{aligned} \tag{3}$$

where, F_i is the fact, m^{t-1} is the previous episodic memory and $m^0 = q$, M_i in this case is 196. Then attention gate is then help to extract a contextual vector c^t base on the attention by using a weighed summantion of facts.

$$c^t = \sum_{i=1}^{N} g_i^t F_i \tag{4}$$

The contextual vector could also be generated by using attention based GRU. however, due to time limit, I was not able to reproduce it(I tried but it takes too long to work and give up). The episodic memory then is updated by using ReLU for its simplicity as well.

$$m^t = ReLU(W^t[m^{t-1}; c^t; q] + b) \tag{5}$$

However, it could also be updated by using GRU as well. It need to be reproduced if more time is allowed.

3.5 Test Models

In this project, apart from two baseline models, pure yes and LSTM question only, I also reproduce CNN + LSTM, and DMN model.

3.5.1 CNN + LSTM

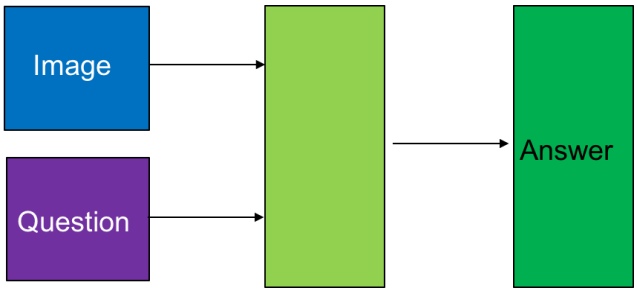

Figure 5: CNN + LSTM Model

For the CNN + LSTM model, I directly concatenate the visual feature extraction module, question understanding module and answer generation module directly to generate answer. The output of visual feature extraction module and question understanding module are directly input to answer generation module.

3.5.2 Dynamic Memory Network(DMN)

Therefore, for DMN model, I only applies weighted sum and ReLU for attention mechanism. The filtered visual feature are concatenated with the question features and input into question answering module directly.

4 Test and Discussion

In this project, I tested several methods:pure yes result, LSTM question, CNN + LSTM, DMN. Among these test, LSTM question and CNN + LSTM are trained by 60 epoches. However, due to

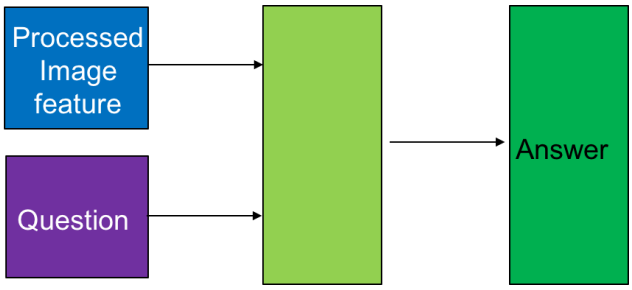

Figure 6: DMN Model

Table 1: VQA Test Questions Accuracy All

Model	Test Accuracy
Pure Yes	27.12
LSTM Question	42.99
CNN+LSTM	51.63
DMN	50.98

time limit, DMN is trained with weigh less epoches. Therefore, the benefit of DMN might not be that obvious. The result is shown as follow: From the result we can see, that for some questions, the system does not always require images. The LSTM question model could generate good result directly from past knowledge. The CNN+LSTM use fully connected later directly as input to the answer module could provide relatively good result, various techiniques, such as normalization, iBowing could help to improve the accuracy in some way. The DMN takes local features and tries to focus on the features related with questions.The tested DMN model performs relatively well with few epoches. I believe if more time is allowed and better computational device could be provided, it will definitely outperform CNN+LSTM model.

For the implementation of DMN, I encountered a lot of problems related with the computation issues such as out of memory, out of hard drive and etc.

Here are some sample outputs of the system and visualQA's interface.

Question: Did a man or woman use the toilet last?
Picture id: 264461
Answer: Man
Human answer: man;man;man;both;man;male;man;man;man;man

Figure 7: sample output

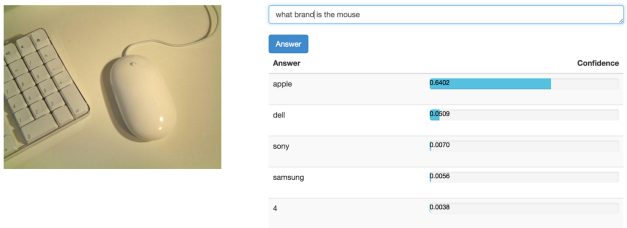

Figure 8: visualqa interface

5 Conclusion and Future Work

In this project, I investigate various methods to deal with visual question answering problem. Based on the impetus of CNN and RNN, I tested four different methods that handles the problem from different perspective. Reasonable results are reproduced in this project. The original purpose of the project was to improve the accuracy. However, due to time limit, I could only reproduce these result. For future work, I will reproduce the DMN model with different attention mechanism and improve the accuracy.

6 Reference

References

[1] Stanislaw Antol et al. "VQA: Visual question answering". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 2425–2433.

[2] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. "DenseCap: Fully Convolutional Localization Networks for Dense Captioning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[3] Andrej Karpathy and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3128–3137.

[4] Ankit Kumar et al. "Ask me anything: Dynamic memory networks for natural language processing". In: *arXiv preprint arXiv:1506.07285* (2015).

[5] Caiming Xiong, Stephen Merity, and Richard Socher. "Dynamic memory networks for visual and textual question answering". In: *arXiv preprint arXiv:1603.01417* (2016).

[6] Bolei Zhou et al. "Simple Baseline for Visual Question Answering". In: *arXiv preprint arXiv:1512.02167* (2015).