Observe, Evaluate, and Discriminate: Visual Question Answering

MedusaLafayetteDecorusScheisse

May 27, 2018

1 Team Information

莊佾霖(b02901026)
Attention Model, Inception Features Extraction

黃意堯(b02901042)
Recurrent Model Trial, Stop Words, SvmLoss

高秉聖(b02901016)
Memory Network, Dynamic Memory Network

方爲(b02901054)
Recurrent Model, Captions in recurrent model, Memory Network on captions, Parsing

2 Introduction

Visual Question Answering (VQA) [1] is a multi-discipline Artificial Intelligence (AI) research problem that combines Computer Vision (CV), Natural Language Processing (NLP), and Knowledge Representation & Reasoning (KR). Research in VQA has dramatically increased in recent years. This article describes our approach to the VQA multiple-choice answering task. In this task, we are given a question and its corresponding image; the system has to predict the answer from one of the five choices. We implemented models using different topologies, ranging from the feed-forward networks to attention-based methods. The detailed procedures are explained in the next section.

3 Approach

3.1 Word Features

We have to represent words with vectors in order to begin our training. Good word representations should certainly lead to good performance. One choice is to use the bag-of-words representation, but high dimension vectors are memory consuming and would require too many parameters. Dimensionality-reduction methods such as principle component analysis (PCA) or Latent Dirichlet Allocation (LDA) should then be applied. Another choice is to convert the words into vectors through unsupervised training. We chose the latter, using publicly available 300-dimensional GloVe vectors [2] pre-trained on 2.2M vocabularies via common crawl.

3.2 Image Features

Image features can be extracted in various ways. Low level processing such as edge detection can be implemented by applying filters. In recent years, convolutional nets have produced state-of-the-art performances in large-scale image and video recognition. Thus we chose to extract image features by conv nets. We used two different state-of-the-art implementations of convolutional neural nets to obtain our features.

3.2.1 VGG

The first pre-trained model we used in out task was developed by Visual Geometry Group (VGG) [3], which applied a very deep network to classifying pictures. The model is readily compatible with Caffe toolbox, and thus extracting features is available by feeding in the training pictures thoroughly. Pre-extracted features are also publicly available.

3.2.2 Google Inception

While VGG features have already achieved high performance in image recognition, Google recently released an even-deeper model named Inception_v3 [4]. Since the topology of the two models are quite different, we decide to use both extracted features in our work by mixing up two of them with some weights. We then observe the performance in our task while tuning the weight.

3.2.3 MSCOCO captions

In addition to using convolution nets to obtain graphical features, we're able to obtain some extra information about the images via image captions, provided by Microsoft: Common Object in Context (MSCOCO) [5]. Each image in our dataset were tagged with four to five sentences that describe some facts about the objects within. By transforming those captions into GloVe vectors, we figure out that the extracted features indeed represent the training pictures adequately.

3.3 Data Preprocessing

3.3.1 Standford Parser

In order to convert words into GloVe vectors, we parsed the question, choices, answers, and image captions with the Stanford Parser [6] to obtain tokens for mapping.

3.3.2 Stop Words

Stop words, which are the most common words, such as ”and” and ”the”, contain little lexical meaning. Filtering stop words can prevent our model from learning about words that are not significant.

3.4 Feed-forward Models

3.4.1 CNN

We would like to know which kind of information has a greater impact on our performance in this task. Therefore we trained our neural models first on the question features and image features separately. Our first model passes the image features extracted from the convolutional nets through a fully connected neural net. The training target is a 300-dimensional GloVe vector. The prediction vector is compared to each given choice, and the one with the highest cosine similarity is chosen as the final predicted answer.

3.4.2 WordVec-CNN

The WordVec-CNN model is based on the previous model, the novel CNN model. Here the words in each question are averaged to obtain a single vector that represents the question. It is concatenated with the image features before being passed through the neural net.

3.5 Recurrent Models

Recurrent models are capable of processing sequences of arbitrary length. Recently, recurrent models have become the mainstream architecture for natural language tasks due to their representational abilities and effectiveness of capturing long-term dependencies. LSTM networks [7] have been successfully applied to a variety of tasks. Our recurrent models for this VQA task uses the LSTM network extensively.

3.5.1 LSTM

LSTM nets are able to capture long-term dependencies in a sentence, therefore it can better represent a sentence. In section 3.4.1, we described a model using only image features. We would also like to know the performance of answering multiple-choice questions on images using only the question information. In this model, the word vectors of a sentence are fed into a LSTM network to get a 300-dimensional vector, and choice-deciding is the same as the previous cases.

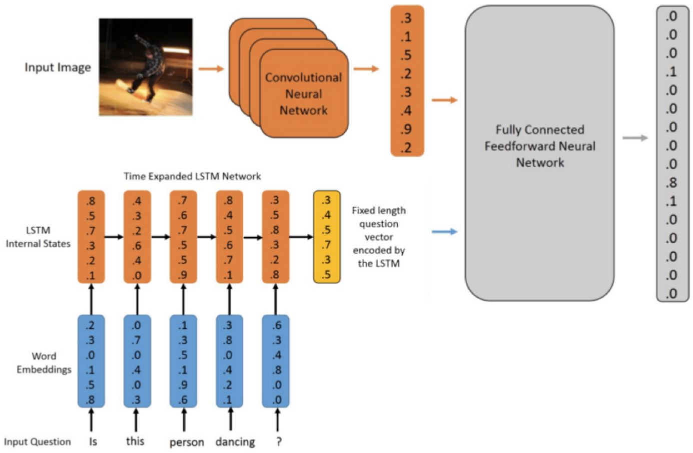

Figure 1: LSTM-CNN model

3.5.2 LSTM-CNN

Since LSTM creates better representation of sentences, we should try adding image features to the LSTM model described above. The output of the LSTM network is concatenated with the image features, and is fed into a feed-forward network. It is similar to the WordVec-CNN model except the average word vector of the sentence is replaced by the LSTM output. The model is shown in Figure 1.

3.5.3 LSTM-LSTM

In section 3.2.3, we mentioned that each image in the MSCOCO dataset is annotated with four to five captions. When the caption sentences are converted into GloVe vectors, we can obtain another image feature. We would like to compare this feature to those extracted from the convolutional nets. The LSTM-LSTM model feeds the word vectors of the caption and sentence parallel through two LSTM nets, and the outputs are concatenated and finally passed through a fully connected net.

3.5.4 LSTM-LSTM-CNN

The LSTM-LSTM-CNN model is a combination of the models mentioned above. The two LSTM nets accept captions and sentences in parallel, and the outputs are concatenated with CNN features before being passed through the neural network. This is the most sophisticated of the recurrent models.

3.6 Attention-based Models: Memory Networks

3.6.1 Attention on Captions

Sukhbaatar et al. proposed the end-to-end Memory Network topology [8], which performs well on natural language problems. In the recurrent models mentioned above, CNN features were used to represent image information. Only one vector is extracted per image in the previous cases, so that attention based methods cannot be implemented. However, if we perform attention on each word in a caption for an image, we will be able to use this topology. The question is passed through a LSTM, with the output being the query q, and q is subsequently embedded to obtain an internal state u. At the same time, each word in the caption is converted into a GloVe vector, and is also embedded into memory vectors m_i. The match, or attention, is computed between u and each memory m_i by taking the inner product followed by a softmax. The attention is then used as weights to extract evidence from another memory embedding, and is summed to get the memory output o. The final word vector prediction is generated by passing the sum of o and u through a fully connected network. Similarity between the prediction and choices are computed like in previous cases. Multiple computational steps, or hops, can be performed by passing the sum of o and u through the memory embedding and making another inference before feeding it through the neural network. The overall model is shown in Figure 2.

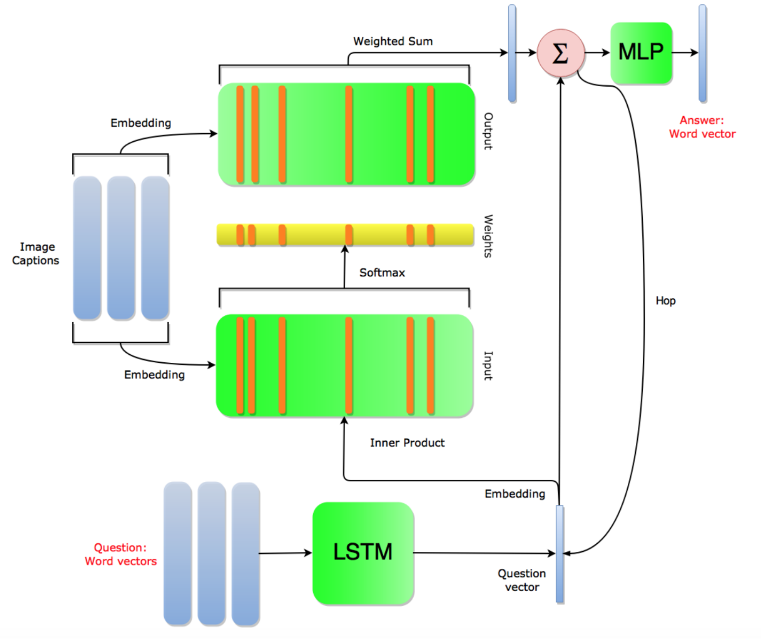

Figure 2: End-to-end Memory Network

3.6.2 Spatial Attention

While the question-driven attention mechanism is performed on embedded memory modules in the original memory network architecture, Huijuan Xu et al. proposed another attention-based model that is trained to focus on the spatial information of images [9]. While single features extracted from the fully connected layer in CNN is used to represent pictures as a whole, the feature vectors generated right after convolutional layers may contain spatial information. Spatial attention mechanism is then performed according to the position of the features. In our task, we implement a sentence-driven attention model by feeding the word vectors of input questions to a LSTM to generate question features. And then sentence attention is calculated by performing sentence feature inner product with the feature vectors distributed on various positions. The sentence attention after a softmax layer now serve as a mask that reveal the relative importance of the spatial features, and thus the attention-based weighted sum of the spatial features along with the question features can be used for inference similar to the previous case. Again, we may also build up multi-hops attention-based model by feeding the output feature vector back to the attention module for another hop. The overall model is shown in Figure 3.

4 Experiments and Results

4.1 Model Details

Recurrent Model:
In pure LSTM model, we implement a three-layer Long-Short-Term network which directly output the inference. As for LSTM-CNN model, a single layer LSTM along with a pre-trained CNN model are concatenated to feed forward to a 3*512 MLP layer, which generates the final inference. LSTM-LSTM model is an end-to-end model that concatenates two single-layer LSTM before the 3*512 MLP layer. Finally, LSTM-LSTM-CNN model is a mixture of the models above.
Attention-based Model:
For the end-to-end memory network model, we tried various embedded dimensions and hops. And the output MLP layer is set to be 2*1024. And for the spatial-attention based model, we set our embedded dimension and hops to 50 and 2, and the MLP layer is set to 2*1024.

4.2 Training Details

Among all of our different trials, we set our learning rate to 0.0002, Keras optimizer to 'RMSProp' and activation function to 'Relu' while training. For models with MLP layer, we also apply dropout to it with dropout rate 0.3. Finally, we've tried several kinds of DNN's loss functions in recurrent models, and they will be listed in the result section.

4.3 Results

Output Activation function	Accuracy(%)
Linear Cosine Similarity(VGG)	78.25
Linear MSE	≈56.923

Table 1: Comparison of output activation function

hop	Accuracy(%)
1	77.56
2	81.629
3	81.212

Table 2: MemNN hop comparison

- **Recurrent Model: Loss function comparison**
  When implementing recurrent model, we encountered the choice of different loss functions, and it turns out that the performance of the best accuracy may vary somehow significantly, which is shown in Table 1.

- **Memory Network: hop**
  We tried 1-hop, 2-hop and 3-hop memory network model in our task, and the result in Table 2 shows that 2-hop model achieves the best performance.

- **Comparison between models**
  After implementing various models, we compare the performance of them according to their best accuracies. And Table 3 shows that MemNN achieves the best performance in the VQA task.

Model	Accuracy(%)		
	VGG	Inception	Combined
CNN	58.14	-	-
WordVec-CNN	69.01	-	-
LSTM	76.43 (No CNN)		
LSTM-LSTM	78.629 (No CNN)		
LSTM-CNN	77.038	78.25	78.810
LSTM-LSTM-CNN	78.418	79.635	79.832
MemNN	81.948 (No CNN)		
Atttention-based NN	-	77.1097	-

Table 3: Comparison between models

5 Conclusion

Visual question answering task is a problem that consists of various AI sub problems, such as computer visual recognition, sentence comprehension and inference problem. As for the visual recognition task, one usually aims to figure out a best feature vector that can represent the corresponded image. Our work shows that even though adopting CNN-based feature vector as input information is seemingly reasonable, picture captions provide us with a better representative vector that describe the image. On the other hand, since GloVe vectors have great performance on the word comprehension task, we try to obtain a proper feature vector that best describe a sentence in our work. The results show that parsing the input sentence properly according to Standford grammar parser before turning them into GloVe vector improve the performance of the entire network. Meanwhile, we do even better by feeding each of the word vectors to a single layer LSTM and taking the final output to be the sentence feature vector. Finally, one simple but straightforward method to tackle with the inference problem is to construct a Multi-Layer Perceptron (MLP) with image and sentence feature vector input, and calculate the loss function with respect to the five different answers. The back propagate mechanism during training may automatically fit the entire model to the desired state. And the result shows that this architecture can provide us with test accuracy at around 78 to 79 percent. However, an even better architecture named attention based model can somehow improve the total performance. The reason is rather obvious: the goal of VQA task is to pick the answer according to both input image and question. Therefore, we can expect that there shall be some correlation between images and sentences, and that's indeed how human solve the VQA task by taking attention to image factors that are important to the question. Both memory network and spatial attention-based model implement similar attention mechanism in the task, and the overall performance achieve test accuracy up to 82 percent. As a conclusion, we may state that attention-based model along with proper input sentence and image features best solve the VQA problem according to our work.

Reference

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh. VQA: Visual Question Answering. International Conference on Computer Vision (ICCV) 2015.

[2] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.

[3] Karen Simonyan, Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. ICLR, 2015.

[4] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. arXiv:1512.00567 [cs.CV]

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár. Microsoft COCO: Common Objects in Context.arXiv:1405.0312 [cs.CV].

[6] Richard Socher, John Bauer, Christopher D. Manning and Andrew Y. Ng. 2013. Parsing With Compositional Vector Grammars. Proceedings of ACL 2013

[7] Sepp Hochreiter, Jürgen Schmidhuber. Long short-term memory.Neural Computation, Volume 9 Issue 9, Nov. 15, 1997, Pages 1735 - 1780

[8] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, Rob Fergus. End-To-End Memory Networks. arXiv:1503.08895 [cs.NE]

[9] Huijuan Xu, Kate Saenko. Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering. arXiv:1511.05234 [cs.CV]