Visual Question Answering

Ashwini Pokle
Department of Computer Science
Stanford University
ashwini1@stanford.edu

Stefanie Anna Baby
Department of Computer Science
Stanford University
stef96@stanford.edu

Abstract

Visual Question Answering (VQA) is a field of research of immense significance that combines natural language processing with computer vision. The task is to develop AI systems that can understand and reply to questions based on a visual input. In this project, we model answering several open ended questions from images given the input text. As our baseline, we measured the performance of combining text and image using an LSTM module with CNN on the Visual Question Answering Dataset (VQA V2). This model by itself gave an accuracy of 40.06% which is solid, given the free-form nature of questions and diverse coverage of the data that was also specifically cleaned to remove language bias. We then implemented Stacked Attention Networks and Dynamic Memory Networks, both of which are stronger models in capturing the semantic representations.

1 Introduction

Answering Visual Questions is a longstanding area of study that has gained huge interest commensurate with the rich set of applications they enable. Notable among its applications is its use in assisting visually impaired individuals to understand contents of images. However, making a machine understand an image is as challenging as it is intriguing. It involves multimodal learning that combines effectively representation of text with that of image. In this project, we look at the results of training a popular but strong baseline for VQA along with improvement that additional model complexities like attention bring in.

2 Background

VQA is one of those tasks that gained popularity after deep learning approaches began improving state-of-the-art performance on vari- ous vision and NLP challenges. Recent years saw the development of several approaches as well as more rigorous evaluation protocols to tackle this appealing intersection. There have also been much effort to develop datasets that remove the strong prior and bias seen in natural language which otherwise allow models to gain good superficial performance without any underlying understanding of the visual content. Visual Question Answering Dataset and Challenge (VQA v2.0) [5] is a relatively new and balanced dataset that has carefully removed several such priors. We have used the same for all our experiments and baseline in this project.

3 Related Work

3.1 Hierarchical Co-Attention

Hierarchical Question-Image Co-Attention is a method that symmetrically attends to the question and image representation where one is used to guide attention in the other [8]. First the question is represented hierarchically at 3 different levels: word, phrase and the level of question itself. These

in turn are used to construct image-question co-attention maps in parallel or alternating fashion. The co-attended features are then recursively combined from word level to question level for the final answer prediction.

From the input question Q word level embedding $Q^w = \{q_1^w, \ldots, q_T^w\}$ are first created. For the phrase features, 1-D convolution of the word embedding vectors is computed using filters of window size unigram, bigram and trigram. Max-pooling across different n-grams at each word location gives the phrase-level features q_p^t. The sequence q_p^t is then encoded using LSTM whose hidden vector at time t gives the question-level feature q_s^t.

Parallel Co-Attention In this scheme, image and question attention are generated simultaneously in each level of the question hierarchy. For this, an affinity matrix C is computed from image and question feature maps V and Q as $C = tanh(Q^T W_b V)$. Then the image (or question) attention is computed by maximizing the affinity over the locations of other modality; i.e, the attention weights are given by $a^v[n] = max_i(C_{i,n})$ and $a^q[t] = max_j(C_{t,j})$. These are used to calculate the image and question attention vectors $\hat{v}$ and $\hat{q}$ as the weighted sum of the image features and question features.

Alternating Co-Attention In the alternating co-attention scheme, at each level of the hierarchy an attention operation takes as input the image (or question) features and the attention guidance derived from question (or image) as inputs, and outputs the attended image (or question) vector.

For predicting the final answer of VQA, the co-attended image and question features from all three levels are passed through separate multi-layer perceptrons to recursively encode the attention features. A softmax of at the final level gives the probability of answers: $h^w = tanh(W_w(\hat{v^w} + \hat{q^w}))$, $h^p = tanh(W_p[(\hat{v^p} + \hat{q^p}), h^w])$, $h^s = tanh(W_s[(\hat{v^s} + \hat{q^s}), h^p])$, $p = softmax(W_h h^s)$

3.2 Multimodal Compact Bilinear pooling

Multimodal Compact Bilinear pooling (MCB) is a method of pooling the visual and textual representations that is more effective and expressive than simple approaches such as element-wise product or sum [4]. Due to their high dimensionality, the outer product of image and question vectors is typically infeasible. MCB approximates this by randomly projecting the image and text representations to a higher dimensional space using Count Sketch [3] and then convolving both vectors efficiently by element-wise product in Fast Fourier Transform (FFT) space. MCB is used to predict answers for the VQA task as well as fine grained locations for visual grounding. Both these problems require finding the most likely answer or location $\hat{a}$ from the network parameters and is obtained by taking argmax over its set of solutions.

Given two vectors $x_1 \in R^{n_1}$ and $x_2 \in R^{n_2}$, MCB learns a linear model of their outer product given by: $z = W[x_1 \otimes x_2]$ Here $x_1 \otimes x_2$ denotes the outer product $x_1 x_2^T$.

For the task of VQA, the query and context image is processed into vectors x_1 and x_2 and passed through MCB. This is followed by an element-wise signed square-root and L_2 normalization which is used for the final prediction using a fully connected layer.

3.3 Multimodal Low-rank Bilinear pooling

Given two input vectors, bilinear pooling provides richer representations than linear models by taking their outer product (or Kroneker product in case of matrices). Since outer product considers all pairwise interactions among given features, such a layer of bilinear pooling can effectively replace the fully-connected layers in neural networks for combining image and text vectors.

Multimodal Low-rank Bilinear (MLB) pooling is a work that parametrizes full bilinear interactions between image and question spaces using Hadamard product to learn their joint representation [6]. To limit the number of free parameters, the output tensor is constrained to be of low rank r. This method outperformed compact bilinear pooling in visual question-answering tasks and gave state-of-the-art results on the VQA dataset (v1). It also has better parsimonious property because calculating the exact expectation over the projected dimensions in compact bilinear pooling is computationally intractable and hence keeps the random parameters in the projections fixed during training and eval-

uation demanding the projected dimensions to be large enough to minimize the bias from using fixed parameters.

3.4 Multimodal Tucker Fusion

Multimodal Tucker Fusion (MUTAN) is another method to efficiently parametrize bilinear interactions between visual and textual representations using multimodal tensor-based Tucker decomposition [2].

While MLB has been very successful in its performance on VQA database, its low rank tensor structure is equivalent to a projection of both visual and question representations into a common space, where simple element-wise product computes the interactions. Thus MLB relies on a simple fusion scheme and learns a mono-modal embedding for text and image. For the VQA task, it is important to merge both modalities by learning very precise correlations in order to decide which answer is correct.

MUTAN does this by multimodal fusion based on bilinear interactions between modalities. Since fully- parametrized bilinear interactions are intractable in VQA due to the prohibitive size of the full tensor, MUTAN controls the number of parameters by reducing the size of the monomodal embeddings, while modelling interaction as accurately as possible. It also does further structuring on the Tucker decomposition of input tensor to improve flexibility over the input/output dimensions and decreasing parameter size.

4 Approach

4.1 Baseline

A simple yet effective baseline for VQA is decoupling learning from the image and text and then producing a joint representation using deep neural networks. We used gated recurrent unit (GRU) model for text processing and combined its features with the processed image features of the same size by element wise multiplication. This combined feature vector was passed through two dense layers and a softmax to get the final probability distribution of the answers. The image feature vectors were themselves prepared by preprocessing the image using VGGNet 19 [9], extracting its $14 \times 14 \times 512$ sized 3D feature maps from the final pooling layer and flattening to a 2d vector using a dense layer.

$$v_{I_i} = VGG\_Net(image_i),\ u_{I_i} = tanh(W_I v_{I_i} + b_I), u_{Q_i} = \overleftrightarrow{GRU}(v_{Q_i})$$

$$u_i = u_{I_i} * u_{Q_i} w_i = tanh(W u_i + b)\, u = softmax(w_i)$$

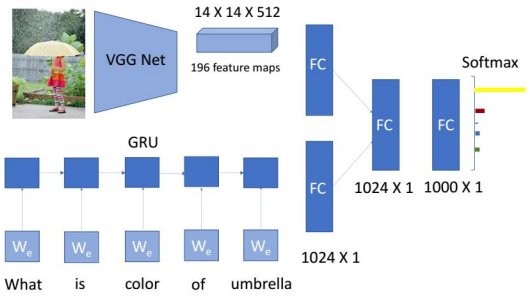

Figure 1: Baseline Architecture

4.2 Stacked Attention Networks

Stacked Attention Networks (SANs) help in answering natural language questions from images using multiple steps of reasoning. SANs extend attention mechanism and uses the semantic representation of a question as query to search for the regions in an image that are related to the answer

[11]. In this method, the image model obtained from a CNN and the question model obtained from a CNN or a LSTM is passed over the stacked attention model that locates via a multi-step reasoning, the image regions that are relevant to the question.

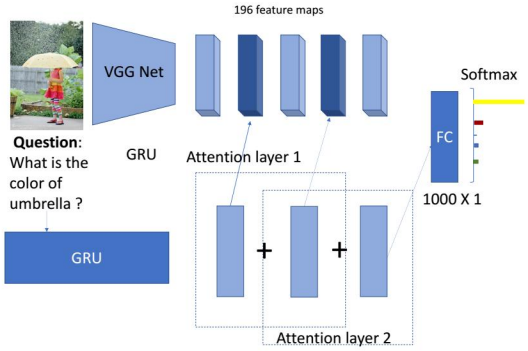

Figure 2: Architecture of Stacked Attention Network

For the image model, we used features from the last pooling layer of VGGNet 19 to extract the image feature map f_I from a raw image I. Hence the image features have a dimension of $512 \times 14 \times 14$, which are then passed through a single layer perceptron to transform each feature vector to a new vector v_I having the same dimension as the question vector v_Q.

Similarly for the question model, we first embed its words to a vector space through using Glove and feed this to an LSTM. Its final hidden layer is taken as question vector v_Q.

Next, the stacked attention module takes in v_Q and v_I and feeds them through through a single layer neural network and a softmax function to generate the attention distribution over the image regions:

$$h_A = tanh(W_{I,A} v_i \otimes (W_{Q,A} v_Q + b_A)$$
$$p_I = softmax(W_P h_A + b_P)$$

Using this attention distribution, we calculate the weighted sum of the image vectors each from a region to get the new image vector $\tilde{v}$. Similarly, the sum of $\tilde{v}$ with the old question vector gives us a refined question vector. In principle, we can repeat this process of fine tuning our vectors using attention over as many steps as we like. The assumption here is that each time our iteration extracts more fine-grained visual attention information from the vectors for answer prediction. But we only used attention for 1 and 2 steps as suggested in the original paper [11]. The final prediction is done via a softmax over the final refined question vector.

4.3 Dynamic Memory Networks +

Dynamic Memory Network (DMN) [7] and its improvement DMN+ [10] is a neural network-based framework trained using triplets of (input text/image, question, answer) to solve textual and visual question answering tasks. The DMN computes a representation for all of the inputs and the question asked, then iterates with attention through the inputs to retrieve relevant facts. The memory module reasons over these facts and generates a vector representation of all relevant information to the question. This representation is then passed to the answer module, which generates the answer.

Visual Input Module : This module encodes input images to extract $14 \times 114 \times 512$ 3D feature maps of the images. These feature maps are projected into linear space and passed through a bidirectional GRU to obtain facts about images which are passed onto the episodic memory module.

$$\overleftrightarrow{f_i} = GRU_{fwd}(\overrightarrow{f_{i-1}}, f_i) + GRU_{bwd}(\overleftarrow{f_{i+1}}, f_i)$$

Question Module : This module encodes questions by passing individual words through a GRU. Only the last hidden state of the GRU is passed to the episodic memory module for further computation.

$$q_t = GRU(q_t, q_{t-1})$$

Episodic Memory Module : This module computes interaction between questions and facts and uses attention GRU to focus on relevant interactions. These interactions are then used to update the internal memory state. Multiple memory passes are performed over the facts and the questions. The final memory state is passed to the answer module for further processing to predict answers.

$$z_t^i = [\overleftrightarrow{f}_i \circ q; \overleftrightarrow{f}_i \circ m^{t-1}; |\overleftrightarrow{f}_i - q|; |\overleftrightarrow{f}_i - m^{t-1}|]$$

$$m^t = GRU(c^t, m^{t-1}),\ g_i^t = softmax(W^{(2)} tanh(W^{(1)} z_t^i + b^{(1)}) + b^{(2)}$$

Answer Module : This module consists of a fully connected layer that takes in question feature and last memory state as input, concatenates them and then predicts answer after a softmax over all possible answer words.

$$y = softmax(W^{(a)} a), a = [q; m_T]$$

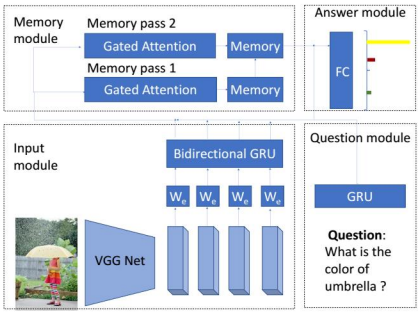

Figure 3: Architecture of Dynamic Memory Networks+

5 Experiments

5.1 Dataset

We used VQA v2 dataset [5] for training models for visual question answering task. The complete VQA v2 dataset contains 443,757 training questions and 82,783 training images source from MSCOCO Image Dataset. The validation dataset contains 214,354 questions and 40,504 images. There are 10 possible answers per question. All the questions are free text and are open-ended; there are not multiple choice type questions. The evaluation metric uses 10 groundtruth answers for each question to compute VQA accuracies. The predicted answer for each question should match with atleast 3 answers to receive full credit. We used the evaluation script provided by VQA to evaluate our models. For all our experiments, we trained on training set and tested on validation set, as VQA does not provide a test set.

In order to reduced language bias, VQA v2 was formed by collecting complementary images such that every question is associated with a pair of similar images in the dataset but results in two different answers. This balanced dataset forces VQA models to focus on visual information and not simply rely on language priors. For instance, in the original VQA data (version 1), the most common answer "tennis" to the question "What sport is" is the correct answer for 41% of the cases, and "2" is the correct answer for 39% of the questions starting with "How many" [1] . Their benchmark results on state-of-art VQA models showed a significant drop in performance, confirming the hypothesis that these models indeed had exploited language biases.

5.2 Input Preprocessing

The images were rescaled to 448×448 and features of the last max-pooling layer of VGG-19 were extracted. Each 3-dimensional volume is a spatial-region map of size $14 \times 14 \times 512$ and there are 196 total maps.

The questions were tokenized with nltk parser. A vocabulary was created from these token. All tokens in the questions were indexed according to this vocabulary. Once indexed, the questions were right aligned and made of same length by prepending zeros.

The answers were also tokenized with nltk-parser. Frequencies of all the answers were counted and only top 1000 answers were considered. A vocabulary was created classes from these answers and the answers were indexed accordingly. Answers that are below this threshold were mapped to none. We framed the visual question answering task as a classification task over these 1000 answers.

5.3 Baseline

We trained and evaluated our baseline on both VQA v1 and VQA v2 to check if there was a significant language prior in given data that our baseline model was exploiting.

5.3.1 Experiments on VQA v1.0

We trained our baseline model with Adam optimizer for 15 epochs. The model started overfitting after 9-10 epochs. We used a batch size of 100 and input embeddings of size 512. We used learning rate of 1e-4. The weights of input embeddings were uniformly initialized between -0.08 and 0.08. We achieved validation/test accuracy of 50.21% on this dataset.

5.3.2 Experiments on VQA v2.0

We used similar experimental configuration and hyperparameters as in the previous experiment. We noted a drop in test accuracy after switching to VQA v2 dataset from 50.21% to 40.06%. This proved that our baseline did not use visual input efficiently and relied mostly on the language to predict answers. We decided to experiment with models that made more effective use of the image features and paid attention to both image and question.

5.4 Stacked Attention Network

Input image features were 196 $3D$ feature maps of size $14 \times 14 \times 512$ which were projected by passing through a fully connected layer to obtain feature matrices. Word features were extracted by passing embedded word vectors through a GRU. The dimension of projected feature space was 512. We experimented with variations of SAN with a single attention layer and two attention layers. We achieved higher accuracy of 47.11% with two attention layers as expected against 44.38% in case of single attention layer. We did not experiment with larger number of attention layers as Yang et. al. [11] report that it degrades performance.

5.5 Dynamic Memory Network (DMN+)

We trained DMN+ on a subset of VQA v2 data of size 100K (roughly 30% of training data) as we were running into out of memory issues while trying to train on the entire VQA v2 dataset. We wrote a disk-based data loader to read a batch of data directly from disk. However, this made the overall algorithm extremely slow and one epoch took around 2.5 hours. Due to computational constraints we trained on a subset of data. We used Adam optimizer with learning rate of 1e-4 and batch size of 100.Size of embedding and the hidden layer was 512. We used three episodic memory passes before predicting the final output. After training on this subset, we achieved accuracy of 21.65% on validation split of the VQA v2.0 dataset. This is lower than its reported state-of-the-art performance, but since the training was only done one-third of actual train data, we couldn't improve any further.

6 Ablative Analysis

We first trained a simple GRU, without any visual input, to predict answers on VQA v2 dataset. This model gave test accuracy of 38.43%. We next introduced a CNN with features extracted from max pooling layer of VGGNet. These extracted features were multiplied with word features extracted from GRU. This model achieved accuracy of 40.06%. We next added attention to our model through Stacked Attention Network which increased our accuracy to 47.11%.

Architecture	Test Accuracy
Baseline-CNN-GRU (VQA v1)	50.21%
Baseline-CNN-GRU (VQA v2)	40.06%
Stacked Attention Network (VQA v2)	47.11%
Dynamic Memory Network *(trained on 100K VQA v2))	21.65%
Hierarchical Co-Attention Network **(reported in [5])	51.88%
Multimodel Compact Bilinear Pooling **(reported in [5])	56.08%

Table 1: Results: Since VQA does not have test split, all reports are on validation split

Answer type	GRU	CNN-GRU	SAN-1	SAN-2
Yes/No	63.69	63.70	63.65	63.52
Number	28.20	30.00	30.33	30.84
Others	23.93	27.90	33.44	36.79

Table 2: Per Answer-Type Performance across models

Architecture	Test Accuracy
Language-based model (GRU without CNN and image input)	38.43%
Baseline-CNN-GRU (VQA v1)	50.21%
Baseline-CNN-GRU (VQA v2)	40.06%
Stacked Attention 1 layer (VQA v2)	44.38%
Stacked Attention Network 2 layers (VQA v2)	47.11%

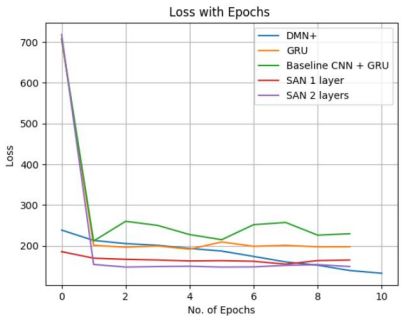

Figure 4: Loss Plots for VQA V2

7 Inference and Conclusion

A simple check of performance on VQA v1 and v2 using our baseline architecture showed a drop of 10% on using v2. The drop is indeed very drastic and demonstrates that the balanced VQA v2 dataset is particularly difficult, requiring VQA models to understand and distinguish between the most subtle differences among the complimentary images in order to predict the answers to both the images correctly. As mentioned by the creators of VQA v2, the complementary images in VQA v2 are close to one another in the semantic (fc7) [5] space of VGGNet features and forces VQA models to learn very careful and differentiating image features to perform well.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

Question type	GRU	CNN-GRU	SAN-1	SAN-2
what	21.32	25.50	28.82	32.28
how	18.62	19.89	17.04	21.36
is	63.89	64.16	63.63	63.34
why	8.03	8.69	9.87	12.24
what is	13.72	18.29	23.66	25.85
what time	15.37	16.21	20.72	21.45
who is	24.05	24.93	17.18	22.89
could	59.72	59.72	59.48	58.17

Table 3: Per Question-Type Performance for selected questions across models

Figure 5: Predictions of 2 layer SAN: How many cats are present in the image? A: 1 (wrong) What type of cat is this ? A: Tabby (right). How many giraffe are pictured? A: 2 (right)

[2] Hedi Ben-younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 1, page 3, 2017.

[3] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming*, pages 693–703. Springer, 2002.

[4] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[5] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, volume 1, page 9, 2017.

[6] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. *arXiv preprint arXiv:1610.04325*, 2016.

[7] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387, 2016.

[8] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016.

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

[10] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *International Conference on Machine Learning*, pages 2397–2406, 2016.

[11] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016.