Visual Question Answering

Pradyumna Gadepally, Karishma Jain, Andrew Wagenmaker, Panfeng Li

I. Introduction

The VQA task [1] seeks to solve the problem of automatically generating answers to questions of images—an important problem in realizing Artificial Intelligence. Human Intelligence can be said to link vision with language in order to create abstract thought. Here, therefore, we try to combine a high-level understanding of images with text. Recently, a balanced VQA dataset has been released [2], overcoming some of shortcomings of the original VQA dataset [1]. This report seeks to address the VQA problem utilizing this new dataset.

We attempted three distinct approaches to the VQA problem. We first cast the VQA challenge as a classification problem—"classifying" the question/image pair into an answer class by using a relatively experimental attempt at applying a Generative Adversarial Network [3] to this classification problem—"generating" answers conditioned on the images and questions. Specifically, we use a Conditional GAN setting. We are unaware of any other attempts at applying GANs in this way and so we sought to investigate their applicability to this problem.

We also attempted two more conventional approaches to the VQA problem. The first one of these employed an autoencoder to learn how the embeddings of the images and questions should be combined to produce an answer. To the best of our knowledge this method has not been attempted for the VQA problem. Our second approach utilized a modified co-attention mechanism to solve the language priors problem. While this approach has been attempted for the VQA problem before, we combined it with another existing approach in a novel way.

II. Generative Adversarial Networks

Our rationale for applying Generative Adversarial Networks to our problem was three-fold:

1) VQA is fundamentally a generative task—given an image and question, we must generate an answer. We thus reasoned that GANs, due to their generative nature, may be a good fit for the problem.
2) Reed *et. al.*'s conditional GAN [4] made us reason that, by inputing question and image embeddings as 2-tuples to the generator, we could learn to accurately generate answer embeddings as the outputs. This is because the generator will learn the probability distribution of the answer embeddings conditioned on the question-image embedding 2-tuple. By thinking of the discriminator term as an adaptive loss function, we also reasoned that, with proper training, this may be able to determine a more appropriate loss for the VQA classification problem.
3) To our knowledge, GANs have not been applied to a problem like VQA—learning a conditional distribution to solve a classification problem. We wished to investigate the possibility of applying this framework to such a problem.

Since the goal of our GAN-based approach was, in part, to simply investigate the application of GANs to classification problems, we focused more on experimenting with and testing a wide variety of possible applications of GANs to this problem than on beating the state-of-the-art results.

A. Architecture

Similar to most existing approaches to the VQA problem, we first passed the image data through a pre-trained ResNet to generate a feature vector for each image. In parallel, we also pre-processed the questions and pass them through an RNN in order to obtain feature vectors for each question. While we do not train the ResNet, we do train the RNN while training the GAN. After generating the feature vectors, we encode them together into a single vector. We attempted two approaches to combining them. Our first approach, motivated by [5], first passed the image and question embeddings through separate $tanh$ layers, then performed element-wise multiplication of these output vectors, and concatenated them with a vector of the element-wise attention of the outputs of the $tanh$ layers. This vector is finally passed through another $tanh$ nonlinearity. Our second approach was much simpler, utilizing separate $tanh$ layers to encode the image and question feature vectors and then combining these embeddings by element-wise multiplication.

Once we had generated the combined embeddings of the features, we fed them into our generator network. These feature vectors can be thought of as the conditional input to the generator, allowing it to generate an output conditioned on the image and question. In addition to feeding in the feature embeddings as conditions for the output, we also inputed random noise into the generator. We tested doing this in three different ways—by concatenating the noise to the feature vector (**N1**), by adding the noise to the feature vector to essentially create a non-zero mean noise input (**N2**), and by feeding in only the features with no noise (**N0**). While the latter approach may not be a true GAN, it can nonetheless be trained as a GAN and gave us a baseline to compare against. We tested with two different architectures for the generator. The first utilized three fully connected layers with ReLu nonlinearities followed by a linear fully connected layer. The second simply utilized a single linear layer. Each of the generators output a vector of length 1000 encoding the likelihood of each of the 1000 most common answers.

We coupled the simple, single layer generator with the simpler of our embedding methods—for future reference we denote this network as $\mathbf{G_{simp}}$—and the more complex generator with our more complex embedding method—which we denote as $\mathbf{G_{full}}$.

We tested with training generators without the discriminator portion and also, in order to produce the full GAN, we fed the generators' outputs into a discriminator network. We attempted several different architectures of the discriminator portion but each was ultimately several fully connected ReLu layers which output a single number into a sigmoid activation to scale it between 0 and 1. In addition to taking the output of the generator as an input, we also fed the features associated with each question and image into the discriminator by concatenating them to the output produced by the generator. We attempted several variations on this, testing with feeding in the raw features produced by the ResNet and RNN and feeding in combined, embedded features. We denote the coupling of the simple generator with a discriminator as $\mathbf{GAN_{simp}}$ and the coupling of the more complex generator with a discriminator as $\mathbf{GAN_{full}}$ A diagram of this architecture is given in Figure 1.

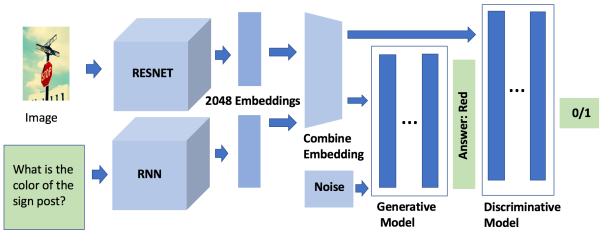


Fig. 1: High Level Architecture of GAN Based System

B. Training

Our basic training algorithm followed the GAN-CLS with step size α, outlined by Reed *et. al.*:

```
Input: minibatch answer embeddings $x$, matching image-question embedding 2-tuple $\psi(t)$, number of training batch step $S$
for $n = 1$ to $S$ do
    $h \leftarrow \psi(t)$ (Encode image-question tuple that matches with the answer)
    $\hat{h} \leftarrow \psi(\hat{t})$ (Encode image-question tuple that doesn't match with the answer)
    $z \sim \mathcal{N}(0,1)^Z$ (Draw random noise)
    $\hat{x} \leftarrow G(z,h)$ (Forward through Generator)
    $s_r \leftarrow D(x,h)$ (correct answer, correct tuple)
    $s_w \leftarrow D(x,\hat{h})$ (correct answer, wrong tuple)
    $s_f \leftarrow D(\hat{x},h)$ (output of Generator, correct tuple)
    $\mathcal{L}_D \leftarrow log(s_r) + \frac{1}{2}(log(1-s_w) + log(1-s_f))$
    $D \leftarrow D - \alpha\frac{\delta\mathcal{L}_D}{\delta D}$ (Update Discriminator)
    $\mathcal{L}_G \leftarrow log(s_f)$
    $G \leftarrow G - \alpha\frac{\delta\mathcal{L}_G}{\delta G}$ (Update Generator)
end for
```

We attempted several variations of this. We experimented with pre-training both the Discriminator and the Generator. When pre-training the Generator we simply trained it as a softmax classifier with noise added to the inputs. This allows us to initialize the weights to values that will produce relatively good results at the start of training the full GAN.

For the Discriminator, we found that pre-training it to optimality is detrimental to the actual training process of the GAN and could worsen the updates of the Generator over time [6]. [6] show that if the probability densities are either disjoint or lie on a low-dimensional manifold then the Discriminator can distinguish between them perfectly. This happens to be the case for the loss function proposed by [3]. So instead of pre-training our Discriminator to optimality, we follow the suggestion in [6] and add noise to its inputs.

We tested adding normalization to the outputs of the layers in our generator and discriminator modules. In general, however, we found that this yielded worse results than when using unnormalized layers. We utilized a small amount of dropout in training our generator and discriminator modules.

We attempted initializing the weights under several different distributions, noticing altered results when we did so. We first initialized all weights to follow a Gaussian distribution with large values clipped (we denote this initialization method **I1**) and also tested with initializing weights to follow a uniform distribution (which we denote **I2**).

III. Attention Based Mechanism

To answer a question according to an image, it is critical to modeling both "where to look" and model "what words to listen to", namely visual attention and question attention [7].

However, [2] shows that the language priors make the VQA dataset [1] unbalanced, where by simply answering "tennis" and "2" will achieve 41% and 39% accuracy for the two types of questions, "What sport is" and "How many". These language priors doubt the reality of whether machines truly understand the questions and images or they only tend to give an answer which has higher frequency in the dataset.

Inspired by the strength of Multimodal Compact Bilinear Pooling (MCB) at efficiently and expressively combining multimodal features [8], we use the MCB operation to replace the simple addition operation used in co-attention mechanism [7] when combining the features learned from the images and questions together, which may help to learn more information from visual part.

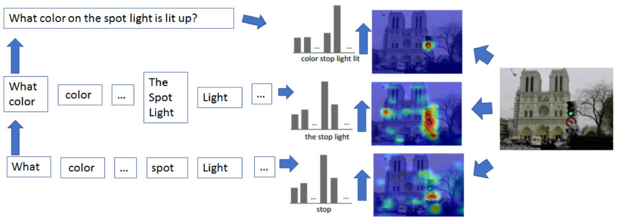


Fig. 2: High Level Architecture of Attention Based System

IV. AutoEncoder Based Mechanism

We modified the initial GAN technique with Autoencoder based technique, wherein the images are passed through a Resnet and Question-Answers through RNN. We concatenate the features and pass it through an autoencoder to generate low dimensional embeddings. Most existing approaches (such as MCB [8]) utilize a fixed method to embed the question and image features together. By employing an autoencoder, we hoped to learn how to best embed the question and image features into a lower dimension space. We finally use this encoding, followed by several fully connected layers, to generate the answer.

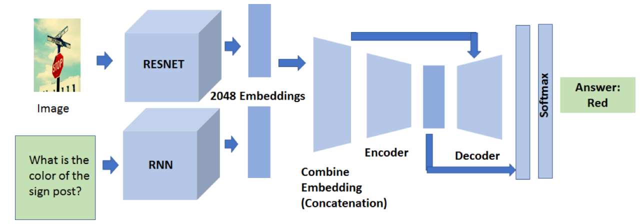


Image
RESNET
2048 Embeddings
What is the color of the sign post?
RNN
Combine Embedding (Concatenation)
Encoder
Decoder
Softmax
Answer: Red

Fig. 3: High Level Architecture of AutoEncoder Based System

Method	All	Yes/No	Number	Other
Baseline Methods				
$\mathbf{G_{simp} - N0 - I1}$	42.70	**67.60**	31.47	26.72
$\mathbf{G_{full} - N0 - I1}$	22.78	56.67	9.05	0.61
$\mathbf{G_{full} - N1 - I1}$	22.48	51.90	22.41	0.08
$\mathbf{G_{full} - N2 - I1}$	22.99	57.28	9.13	0.54
$\mathbf{G_{full} - N2 - I2}$	35.18	63.38	27.36	15.77
Attention	**47.49**	66.64	**32.15**	**36.98**
Novel Methods				
$\mathbf{GAN_{simp} - N0 - I1}$	23.86	62.49	0.25	0.71
$\mathbf{GAN_{simp} - N0 - I2}$	16.23	40.56	5.32	0.59
$\mathbf{GAN_{full} - N0 - I1}$	21.60	55.46	3.49	0.61
$\mathbf{GAN_{full} - N1 - I1}$	13.74	35.77	1.25	0.27
$\mathbf{GAN_{full} - N2 - I1}$	21.25	54.65	3.50	0.51
Attention + MCB	3.36	6.92	0.23	1.48
Autoencoder				

TABLE I: Results on VQA 1.9 Validation Dataset. Legend: $\mathbf{G_{simp}}$ - simple, single-layer generator trained as classifier; $\mathbf{G_{full}}$ - full, multi-layer generator trained as classifier; $\mathbf{GAN_{simp}}$ - simple, single-layer generator trained with discriminator; $\mathbf{GAN_{full}}$ - full, multi-layer generator trained with discriminator; **N1** - noise concatenated to generator conditioning input; **N2** - noise added to generator condition input; **N0** - no noise inputed to generator; **I1** - weights initialized via Gaussian distribution; **I2** - weights initialized via uniform distribution

V. Results

Figure 4 illustrates the effect of adding a discriminator to a generator after pretraining. Here we consider the full GAN model ($\mathbf{GAN_{full}}$), with noise added to the conditioning generator input (**N2**), and weights initialized to a uniform distribution (**I2**). We first train the generator individually as simply a softmax classifier, the accuracy of which is shown in green. At iteration 11000, we add a discriminator module to the end of the generator, optimizing now with respect to the GAN loss. These results are shown in blue. After adding the discriminator, we also continue training the generator individually to compare. As the plot illustrates, adding the discriminator initially hurts the classification results by a relatively significant margin, decreasing accuracy by roughly 10%. This was a behavior we generally observed—the discriminator does not seem to improve the ability of the generator to produce accurate answers. While in this case we did not have time to run each to convergence, even in cases when models were run to convergence, the generator nearly always still ultimately decreased in accuracy when trained with the discriminator.

Figure 5 shows the plot of loss and accuracy versus iteration when training Attention and Attention + MCB model. We can see that the losses of both two models decline to a relatively small value quickly at the beginning, then start to vibrate down. The accuracy has a similar situation with the loss except for the tendency. For the Attention model, the accuracy increases about 0.1 after 6,000 iterations in the end; whereas for the Attention + MCB model, the accuracy increases more slowly. In [2], the Attention model reaches an accuracy of 51.02 on VQA dataset v1.9. And after iteration of 54,000 from raw, our attention model reaches the accuracy of 39.06. We can also find that by adding the MCB part, it doesn't improve the performance at all. On the contrary, the accuracy decrease from 39.06 to 32.03.

Table I gives the numerical results obtained by every method we tested. The metric used is the one presented in [1] where an answer is considered correct and given a score of one if at least three of the ten human given responses match that answer. As this table illustrates, the baseline methods in general do better than all novel approaches we attempted.

Figure 6 illustrates the qualitative performance of a GAN based approach and attention based model on several images. While the GAN is able to correctly answer the simpler questions, it fails miserably at more complex questions such as (c). For (d) it seems to capture some of the meaning of the question (relating a fridge to a meal) yet still incorrectly answers the prompt.

Table II illustrates the effect various pretraining combinations have on the results produced by the GAN. There results are given on the GAN utilizing the full, multi-layer generator model ($\mathbf{GAN_{full}}$) with noise added to the conditioning inputs (**N2**) and weights initialized according to a Gaussian distribution (**I1**). From this it is clear that the optimal results are obtained when the generator is pretrained but the discriminator is not. When we do not pretrain the generator at all, we found that, given the amount of time for which we trained, the generator is unable to learn the distribution of the data at all.

VI. Difficulties Encountered

The primary difficulty we encountered was in training our GANs. While we were able to generate relatively good results when training our generators individually, in general these results did not hold when training a full GAN. We

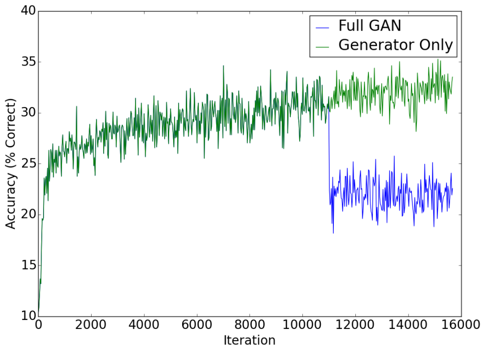


Fig. 4: Effect of Adding Discriminator to Pre-trained Generator

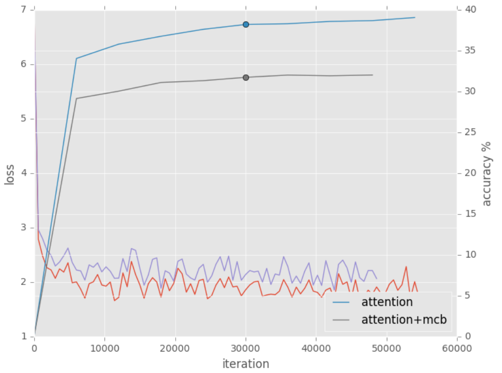
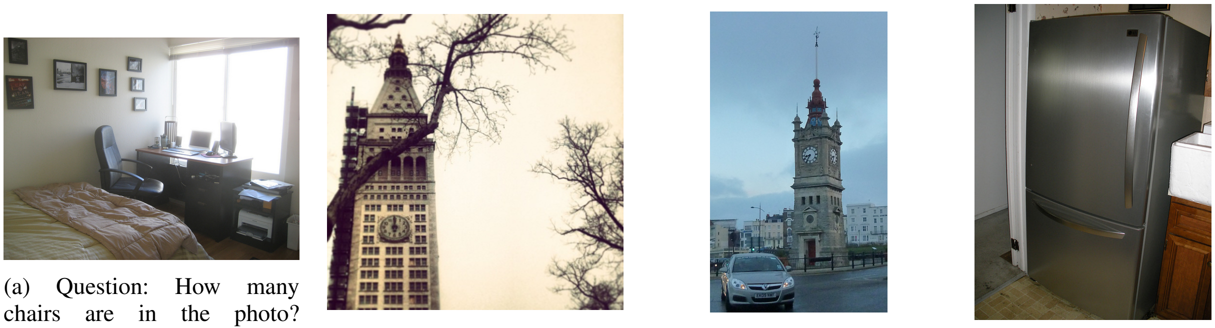


Fig. 5: Training Loss and Accuracy of Attention and Attention + MCB Models



(a) Question: How many chairs are in the photo? Correct Answer: 1. GAN Answer: 1. Attention Answer: 3

(b) Question: Is it an overcast day? Correct Answer: yes. GAN Answer: yes. Attention Answer: yes

(c) Question: What year is the car? Correct Answer: 2010. GAN Answer: scarf. Attention Answer: 2010

(d) Question: What color is the fridge? Correct Answer: silver. GAN Answer: dinner. Attention Answer: silver

Fig. 6: Qualitative Results of Visual Question Answering

Method	All	Yes/No	Number	Other
Neither Pretrained	0.10	0.08	0.19	0.10
G Pretrained	**21.25**	**54.65**	**3.50**	**0.51**
D Pretrained	0.05	0.00	0.09	0.08
Both Pretrained	13.76	35.47	2.16	0.30

TABLE II: Varying Pretraining on GAN with **N2** (noise added) on VQA 1.9 Validation Dataset

surmise that this is primarily due to the problem space being considered. In general, we have seen that GANs are applied to problems where the output space of the generator allows for some ambiguity—for instance, there is not a single correct image that must be produced by a GAN when synthesizing images from text descriptions, a wide variety of images would be deemed correct. However, in our case, we would only accept a single answer as correct and there was thus little room for variation in the output of the generator. We hypothesis that GANs, while applicable to more ambiguous problems, are not well-suited to solve problems such as this where such specific results are desired.

We also encountered other issues simply in the process of training our networks. Due to the activation functions required to produce outputs in our desired ranges, we had difficulty keeping outputs from saturating and gradients from vanishing.

VII. Reflection

In training GANs two functions are learned—a generator to come up with answers and a discriminator to determine whether answers are good or bad. For the scope of this project, we only utilized the generator to determine answers to our questions. However, one could also utilize the discriminator to perform this task by feeding the question and image embedding into the discriminator along with every possible answer choice. The answer that yields the greatest output value from the discriminator should then be considered the correct answer since it would then correspond to the answer most likely to exist. Due to time constraints, we did not investigate this direction but this could be an interesting avenue of work. Moreover, in [6], we learnt that there are problems with pre-training the GAN and its stability with the loss functions proposed in [3]. To this end, they motivate and cite a metric called the Wasserstein-1 which is used in [9]. Since one of the problems we had with pre-training the Discriminator was trying to figure out an appropriate way of bringing in noise into the picture, we would have liked to have tried Wasserstein-GANs.

REFERENCES

[1] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C. and Parikh, D., 2015. Vqa: Visual question answering. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2425-2433).

[2] Goyal, Y., Khot, T., Summers-Stay, D., Batra, D. and Parikh, D., 2016. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. arXiv preprint arXiv:1612.00837.

[3] Radford, A., Metz, L. and Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

[4] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B. and Lee, H., 2016, May. Generative adversarial text to image synthesis. In Proceedings of The 33rd International Conference on Machine Learning (Vol. 3).

[5] Saito, K., Shin, A., Ushiku, Y. and Harada, T., 2016. Dualnet: Domain-invariant network for visual question answering. arXiv preprint arXiv:1606.06108.

[6] Arjovsky, M. and Bottou, L., 2017. Towards principled methods for training generative adversarial networks. In NIPS 2016 Workshop on Adversarial Training. In review for ICLR (Vol. 2016).

[7] Lu, J., Yang, J., Batra, D. and Parikh, D., 2016. Hierarchical question-image co-attention for visual question answering. In Advances In Neural Information Processing Systems (pp. 289-297).

[8] Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T. and Rohrbach, M., 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. arXiv preprint arXiv:1606.01847.

[9] Arjovsky, M., Chintala, S. and Bottou, L., 2017. Wasserstein gan. arXiv preprint arXiv:1701.07875.