Visual Question Answering using Deep Neural Networks

Keerthana Sugasi (ksugasi), Madhavan K R(madhkr), Zachary James Petroff(zpetroff)

***Abstract*—Given an image and a natural language question about the image, Visual Question Answering (VQA)[1] is the task of providing an accurate natural language answer. This task has wide-ranging applications, such as assisting the visually impaired, translating text into images from different languages, etc. The VQA tasks can be of various forms, such as binary visual questions, fill-in-the-blank templates, etc. We attempt to achieve this task in this project using Deep Neural Networks. There are many challenges and subtleties related to this task. The job of the network here is multi-pronged. First, it must understand the objects and their placement in the image. In other words, the network has to identify many objects and actions, the location of the objects in the image, the number of objects and their associations, etc. Second, the network must also develop a good grasp of language or multiple languages. And finally, the network must be able to correlate its findings from the image to its learning in the language to generate a cohesive output in the expected language. We present a modular network architecture using RNN based network for language tasks and Convolutional auto-encoders for vision tasks that result in quality answer predictions.**

I. Introduction

Visual Question Answering(VQA) is a prevalent task for humans to show an image and answer various questions about the image. The questions can be of various forms, such as truth-based questions, for which the answer can be simple Yes/No, and word-based questions, such as identifying objects and their characteristics. While the task of VQA seems naive for humans, it is exciting and challenging to achieve for computers.

The enormous potential this task has, and numerous areas of application for it have attracted machine learning enthusiasts to focus their research in this area in recent years. While the task seems easy, the multi-pronged nature of the task is what poses a challenge to machine learning algorithms. VQA consists of two parts which provide the entirety of context to the task: the image and a natural language question. When we look at these two parts in isolation, the answers and insights one can derive can be completely different than when we look at them together. In other words, one part provides context to the another.

Significant progress has been made in the areas of machine learning, such as Computer Vision and Natural Language Processing. While the former focuses on teaching computers how to understand an image, the latter focuses on teaching computers to discern human languages. VQA provides a unique opportunity that intersects both these machine learning domains. NLP techniques are necessary to understand the natural language question, and the insights derived from the NLP techniques become crucial to derive appropriate insights from the associated image.

A similar but easier task to VQA is a Textual QA, where a natural language question is purely answered from information derived from a large text. Tasks such as reading comprehension and information retrieval are examples of Textual QA. Textual QA tasks fall strictly under the domain of Natural Language Processing and have been studied for a long time. VQA is a much more complex and advanced extension of Textual QA, where an image provides the context for the question rather than a large text. The complexity of the task is due to several reasons (1) Images are higher dimensional data compared to text, (2) Images capture more concrete information than text, and (3) Images lack the structure that texts have.

Image captioning is another task very close to VQA, where a natural language description of the image is generated. Image captioning is limited to extracting and summarizing only the information contained within the image. Whereas, in VQA, the machine learning models often require information beyond what is available in the image to answer a question about a particular image. Hence, VQA is a slightly more complex problem than image captioning.

The approaches to addressing VQA tasks have evolved in the last few years but can majorly be categorized into four major groups[2] (1) **Joint Embedding Approaches** that implement convolutional and recurrent neural networks (CNNs and RNNs) to learn embedding of images and natural language in a common feature space. (2) **Attention Mechanisms** improve on Joint Embedding Approaches to focus on specific parts of the image or question at hand, (3) **Compositional Models** allow building tailored models to address different aspects of the problem and combine them to form a single neural network that reflects the structure of the problem, and (4) **Knowledge base-enhanced approaches** address the use of external data by querying structured Knowledge that is not available in commonly used VQA datasets.

II. Related Work

In this section we discuss two aspects of VQA: The Datasets available for VQA and the Methods developed over last few years to address the task.

A. Datasets

A VQA dataset is made up of three components - an image, a related question, and a corresponding answer - these three components form a single data point. It is important to note that a single image can have multiple question-answer pairs

associated with it. For example, an image of a group of people can have many associated questions such as "how many people are in the photo?", "what is the color of the background?", etc. Hence, the number of instances in a dataset depends not only on the number of images, but mainly on the number of questions each image has. Roughly, the size of a dataset can be inferred as $N \times Q_a$ where N is the number of images and Q_a is the average questions per image. In addition to this, the complexity of the dataset is determined by the length of the questions and answers.

Table 1.[3] provides a list of most commonly used datasets in training and testing VQA models.

TABLE I: Commonly used Datasets for VQA.

Dataset	#Images	#Questions	#Avg. Questions
DAQUAR	1449	12468	8.60
Visual7W	47300	327939	6.93
Visual Madlibs	10738	360001	33.52
FM-IQA	158392	316193	1.99
COCO-QA	117684	117684	1
VQA(COCO)	158392	316193	1.99
VQA(Abstract)	50000	150000	3

*1) **DAQUAR**[4]:* The DAtaset for QUestion Answering on Real world (DAQUAR) is the first and benchmarked dataset for the VQA task. The dataset is a collection of indoor images with semantic segmentation where each pixel represents an object class(total of 894 classes). The question and answer pairs are generated using two ways: 1) Using question templates and 2) Using Human Annotations where the answers were generally one word like color, number of classes etc. The final dataset resulted consisted of 12468 question-answer pairs. The evaluation metrics used for DAQUAR are 1) Simple Accuracy - which is not a good metric for multi-word answers and 2) WUPS - A score between 0.0 to 1.0 (threshold 0.9) based on an average match between answer predicted and ground truth answers.

*2) **Visual7W**[5]:* Visual7W dataset is a collection of 47300 images from MS-COCO dataset and 327939 questions used for image captioning, recognition and segmentation. The 7W in the name gets its significance from the generation of Multiple Choice Questions(MCQs) from interrogative words like Who, What, When, Why, How, and Which. Amazon Mechanical Truk(AMT) workers were used to generate MCQs by drawing bounding boxes of objects for questions in the images to resolve textual ambiguity and to enable the visual nature of the answers.

*3) **Visual Madlibs**[6]:* Visual Madlibs is a collection of 10738 images from MS-COCO dataset and 360001 fill-in-the-blanks and Multiple choice questions. The fill-in-the-blank questions were descriptive and automatically generated using templates and object information. The answers which were words or phrases were formed by AMT workers. The MCQs were evaluated using the accuracy metric.

*4) **FM-IQA**[7]:* The Freestyle Multilingual Image Question Answering (FM-IQA) dataset is a collection of 158392 images from MS-COCO dataset and 316193 questions for which the answers are generated using Baidu Crowdsourcing server. The Question-answer pairs could be words, phrases or full sentences available in English and Chinese. But the dataset required human evaluation using the visual Turing Test which hindered the popularity of the dataset.

*5) **COCO-QA**[8]:* The COCO-QA dataset is a collection of 123287 images from MS-COCO dataset and contains one question-answer per image. The dataset is broadly divided into four categories 1) Object 2) Number 3) Color and 4) Location. Evaluation is done using the Accuracy metric or WUPS score.

B. Methods

The advancements in deep learning models in computer vision and natural language processing have resulted in most of the latest work related to VQA having solutions designed around neural networks. In this section we describe some of the categories of approaches used for solving VQA tasks[2].

*1) **Non Neural Network approaches**:* The non-neural network approach generally takes a probabilistic approach to answer the questions. Multi-World QA[4] models VQA as a probability distribution over answers given an image and a question, i.e., $P(A = a \mid Q, W)$, where A is the answer, Q is the question, and W is the world represented by an image. The above probability can be expanded in terms of a latent variable T corresponding to the semantic tree obtained by parsing the natural language query Q. In other words, $P(A = a \mid Q, W) = \Sigma_T P(A = a \mid T, W) P(T \mid Q)$. This expansion is crucial as $P(A = a \mid T, W)$ is relatively easy to evaluate using a deterministic function. The above model can also be extended to a multi-world setup where the probability distribution of the answer is expanded over values of W.

Answer Type Prediction [9] proposes a Bayesian framework to initially predict the type of answer and then use it to generate an appropriate answer. The VQA task is modelled as the probability of an answer and answer type, given the image and the natural language question, i.e., $P(A = a, T = t \mid x, q)$ where A and T are the answer and answer type, respectively, x and q are image and natural language question respectively. The above probability is then marginalized over all answer types. Finally, the three probabilities are modelled separately, with the first as a conditional multivariate gaussian, the second and the third using logistic regression.

*2) **Joint Embedding approaches**:* The concept of jointly embedding text and images allows one to learn the representation in a common feature space. In the common feature space, image representations are obtained using pre-trained Convolutional Neural Networks (CNNs). Textual representations are obtained from pre-trained word embedding of large text corpora. The word embedding is then fed into a recurrent neural network to capture the syntactic patterns.

A method using a Recurrent Neural Network (RNN) implemented with Long Short-Term Memory (LSTM) cells is suggested in "Neural-Image-QA"[10]. The RNN generates the inputs (Questions) and outputs (Answers), while a pre-trained CNN generates the visual features for object recognition. The

first LSTM layer - encoder receives the query and image features, and creates a fixed-length feature vector that is delivered to the second LSTM layer - decoder. Until a predicted word has a specific END sign, the decoder generates variable-length replies at each iteration and refeeds them into the recurrent LSTM layer. The de facto baseline dataset for VQA is the Neural Image-QA. The authors in [8] proposed a variant of the above approach which uses a VIS+LSTM that passes the feature vector directly into the classifier instead of a decoder to produce single-word answers from the predefined library. This variation converted the approach from sequence generation to a classification problem. Another variation was to use 2VIS+LSTM architecture which takes two image input vectors one at the front and the other at the back. The bidirectional LSTMs better capture the relation between the distant words in the questions by scan questions in both forward and backward positions.

A different method called "Multimodal QA" (mQA) was proposed by work in [7] which employs LSTM to encode and decode questions. This method differs from Neural Image-QA in the following ways 1) common shared weights are used which helps mQA to learn parameters distinctly and 2) the CNN features are used as image representation but are fed into the encoder at every time step not only prior to the question. The method was not publicly tested on datasets unfortunately so it is not comparable.

*3) **Attention mechanisms**:* Attention mechanisms use local image features and assign different importance to different regions to improve the performance of models. For example, in the image captioning tasks, an attention-based model extracts salient features of the image first and then focuses on creating the caption based on those salient features. In other words, the general idea of attention mechanisms is to direct the models on "where to look."

[5] proposes a word-guided attention model based on LSTM. The attention model is introduced by the term z_t, a weighted average of convolutional features that depends on previous hidden states and convolutional features. The attention term controls the contribution of each convolutional feature; a higher value indicates more relevance to a particular portion of the question, and a lower value indicates lower relevance. [11] proposes a "question-guided attention map" that searches the visual features based on the semantics of the input question. [12] proposes a "multi-hop image attention scheme" that combines both the word-guided and question-guided mechanisms in an iterative manner. While the above methods discuss attention models that primarily focus on the language aspect of the data, [13] proposes a co-attention model that aims to jointly reason about image and question symmetrically.

Attention mechanisms have shown significant improvement over models that use global image features. The attention enhanced mechanism by [5] model outperforms the "VIS+LSTM" model for the Visual7W dataset. Fortunately, attention mechanisms have been shown to improve the overall accuracy of all VQA datasets but don't have a major impact when inspected for binary questions. There are a few limitations encountered like 1) The output at the end is end-to-end joint embedding approaches regardless of the attention model used. This leads to no/less insight into how an output answer arises. 2) If an answer can be generated for questions from the given visual input alone. 3) If prior knowledge is required for the previous scenario then in what ways can it be implemented.

*4) **Compositional models**:* Most models discussed above are traditional monolithic networks. Compositional models propose modular network architecture for the VQA task, which involves connecting independently developed models that solve specific portions of the problem to generate a final output. Neural Module Networks [14], designed explicitly for VQA, takes advantage of the compositional linguistic structure of the questions to build a modular network structure. [15] proposes a Dynamic Memory Networks(DMN) structure for the VQA task. DMNs are memory-augmented networks that perform read and write operations on an internal representation of the input. These models allow for complex logical reasoning by modeling complex interactions between multiple parts of the data - in the case of VQA, the image, and the question.

The performance of Neural Module Networks is evaluated against VQA dataset and generally outperforms its competitors on questions with a compositional structure. The bottleneck that develops during the question processing is what causes these approaches' inherent limitations. However, the dearth of challenging questions in the VQA benchmark restricts NMN's true potential in actual use. the overall approach of NMN addresses the potential of a combinatorial explosion of concepts and relations arising in the VQA world. The Dynamic memory networks(DMN) were evaluated against the DAQUAR and VQA benchmarks but perform similarly to NMN on the binary answer questions and slightly worse on the numerical types. Here the same model is applied to both VQA and textual QA giving rise to potential criticism which stems from the intrinsically different nature of sequences of words and sequences of image patches.

*5) **Models using external knowledge bases**:* The approaches have focused only on the visual and textual information part of the VQA data. There has been the development of models that rely on additional information, both textual and visual, to generate a better understanding of the inputs and hence to answer. This model decouples the reasoning from the actual storage of data or knowledge. Ahab is a VQA framework that uses large structured knowledge bases to train the model first. Visual features are extracted using CNNs from a given image and associated with the text from the knowledge base. An improvement to this approach uses an LSTM and a data-driven approach to learn the mapping of images/questions to the queries.

Most of the VQA dataset consists of a majority of question which requires prior knowledge and hence performance of these methods reflect poorly on the datasets. The authors have come up with small-scale datasets for which Ahab outperforms the joint embedding methods on the KB-VQA in terms of accuracy and FVQA outperforms the conventional approaches. However, the major limitations of these evaluation methods are less number of question types and small-scale datasets.

III. Proposed work

A. Materials and Methods

*1) **Modular Networks**:* A neural network or machine learning algorithm capable of answering questions based on images must be able to extract both linguistic and visual information. A modular network design is one way to address the need for processing multiple types of information. A modular neural network is made up of several smaller neural network models that solve portions of a problem. They generally have an integrator module responsible for breaking down tasks into smaller parts, assigning them to appropriate network modules, and gluing and integrating the responses to generate a final output. The Modular network helps decompose complex problems into smaller ones by reducing their score and making them easier to solve. The VQA task is a natural fit to be developed as a modular network. In a VQA task, the final model must be able to learn a natural language component and a visual component; these tasks can be done separately in their own modules and easily be integrated to generate a final response. Hence, For this problem, we separated the language and vision tasks. A network capable of handling language is trained separately from a network capable of handling images, then combined to produce a singular output. By doing this, we cannot only fasten the training process but also improve training process by using better data quality. For example, by training the language model separately, we could use rich data to train it, hence getting better insights from the model on the VQA data.

*2) **LSTM**:* A Recurrent Neural Network(RNN) is a class of artificial neural network that allows cyclic connections between nodes, which allows these networks to make use of context to make a prediction. RNNs have proven very useful in applications such as predicting the next word of a sentence. However, RNNs have problems with long-term dependencies[16], i.e., although RNNs can use the previous context to answer current questions, they fail to keep longer-term context. For example, RNNs might do a good job predicting the next word of a sentence based on a single sentence, but they need to perform tasks where they are expected to keep context over several sentences.

Long Short-Term Memory Networks (LSTMs) are RNNs that address the above issue of long-term dependencies. It is designed for remembering long-term memory, so it should be able to consider relationships between distant words, such as a word at the beginning and the sentence's end. Hence, LSTMs would better predict the next word of a sentence using a more extended context than normal RNNs.

*3) **Autoencoders**:* Autoencoders are a special kind of artificial neural network that produces an output that is the same as its input. The networks work in two parts - first, a high-dimensional input is compressed into a lower-dimensional representation, and second, the lower-dimensional code is reconstructed to its original form. It is important to note that the compression is done in a manner such that there is minimal information loss. The autoencoders have two key components

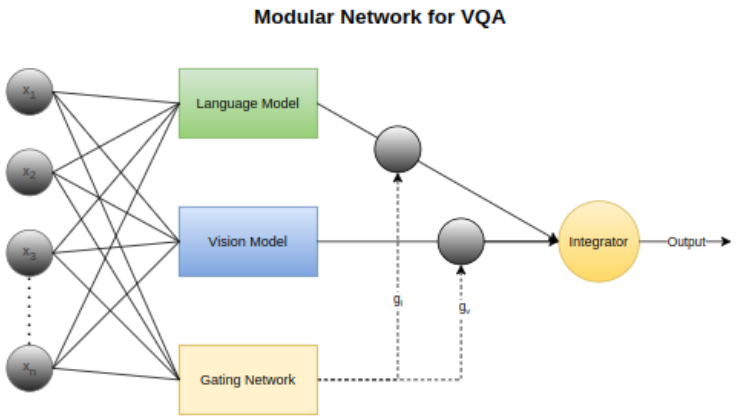


Fig. 1: Modular network architecture for VQA

- (1) Encoder: part of the network that learns how to compress the input, and (2) Decoder: part of the network that learns how to reconstruct the input from the compressed encodings.

Convolutional autoencoders are used to learn convolution filters, generally applied in image reconstruction tasks to minimize errors. These autoencoders can be applied to extract essential features of an image. Convolutional autoencoders are a good fit for image data because they keep the spatial information of the input image and extract information using the convolution layer. The encoder component is made up of convolutional layers that extract features of the image to represent the image in a compressed format. The decoder component is made of transposed convolutional layers (De-convolutional layers) that can reconstruct the original image from the features extracted by the encoder.

B. Overview of network architecture

We have designed our artificial neural network as a Compositional model that is made up of two independent models, one natural language model and one vision model, which are combined to generate output for VQA data. These separate models have been developed as External knowledge-based models, i.e., these models are trained on external datasets. The language model is initially trained on the WikiText dataset, and the vision model is trained on the coco 2014 dataset.

C. Datasets used

As discussed, we have made use of a modular network architecture to solve the VQA task. Two independent models were developed, one for the natural language and one for the visual data, which are then combined to answer the VQA task. Hence, we have made use of 3 datasets - (1) Wikitext dataset for training the language model, (2) COCO 2014 dataset for training the vision model, and (3) VQA V2 dataset for the composite model.

The WikiText[17] language modeling dataset is a collection of over 100 million tokens extracted from the set of verified Good and Featured articles on Wikipedia. COCO [18] is

a large-scale object detection, segmentation, and captioning dataset. Visual Question Answering (VQA) v2.0 [19] is a dataset containing open-ended questions about images, it has 265,016 images, 5.4 questions per image, atleast 10 ground truth answers per question, and 3 plausible but false answers for each question.

D. Language Model

To extract lingual information, a Recurrent Neural Network (RNN) network was built and trained on the wiki-text dataset to predict the next word given a sequence of words. This dataset was chosen because the questions and answers within the VQA dataset themselves do not contain enough information and diversity to allow a neural network to understand language.

The first layer of the RNN is an embedding layer, which transforms the input into a 1024-dimensional feature vector. The embedding layer is then followed by three Long Short-Term Memory (LSTM) layers, each with a hidden state size of 1024. These layers are then followed by a fully connected layer of size 29,473 (representing the size of the network's vocabulary). Dropout (0.65) is used between every layer in the network to prevent over-fitting.

The network is trained with an initial learning rate of 1e3. If the validation loss did not decrease between epochs, the learning rate was reduced by a factor of 0.5. The best model (determined by validation loss) is saved and used for the final model.

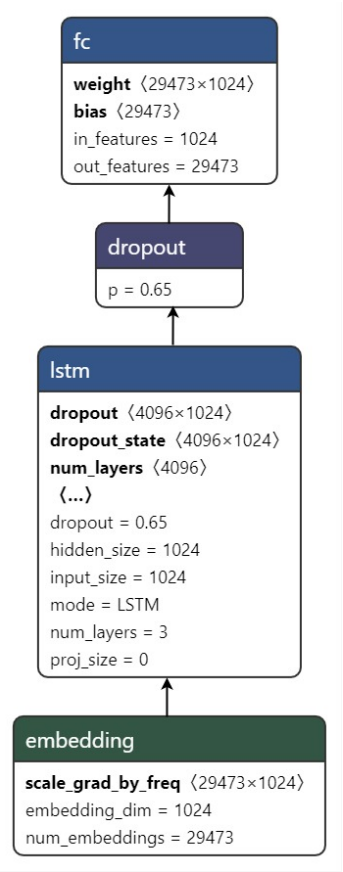


Fig. 2: Network structure for the Language Model

E. Vision Model

To extract visual information, an Auto-encoder was created and trained on the COCO 2014 dataset, consisting of over 330,000 images. The COCO dataset was chosen because the VQA data set was created from images sampled from COCO.

The encoder network consists of four convolution layers, followed by one fully connected layer, of size 1024. The first two layers have a kernel size of three, while the last two layers have a kernel size of five. The stride is set to one for all layers, except for the last layer, where a stride of two is used. Max-pooling is used between the last convolution layer and the fully connected layer to reduce the size of the output vector, while minimizing information loss. Batch normalization and dropout (set to 0.4) are used between every layer to prevent over-fitting.

The decoder network consists of one fully connected layer followed by four deconvolution layers. The fully connected layer was set to size 16,500. The output of the fully connected layer is reshaped into a 100x50x50 tensor before being passed to the first deconvolution layer. The first two deconvolution layers have a kernel size of three, while the last two layers have kernel sizes five and four, respectively. A stride of one is used in the first two deconvolution layers, while a stride of two is used in the last two deconvolution layers. Batch normalization is used between every layer in the decoder network, again, to prevent over-fitting.

Before training, all images were resized to 224x224x3, to make the data set uniform and allow for batching. Every layer in both the encoder and decoder networks utilize the leaky ReLU activation function. The network was trained for eight epochs, with a batch size of sixteen and an initial learning rate of 1e-3. If the validation loss did not decrease between epochs, the learning rate was reduced by a factor of 0.5. The best model (determined by validation loss) is saved and used for the final model.

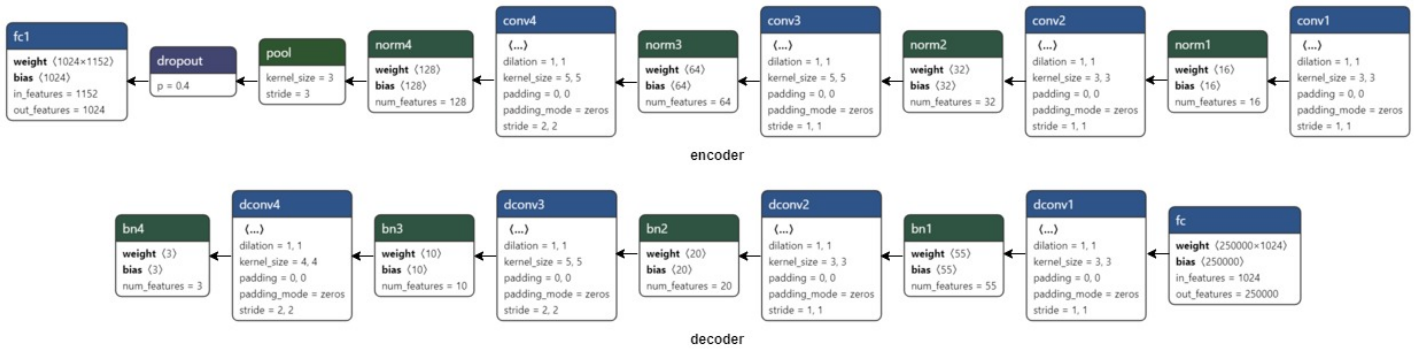
F. Combining the Models

To combine the visual and lingual information, we concatenated the 1024-dimensional output of encoder network and 1024-dimensional embedding output of the RNN to create a 2048-dimensional feature vector. This vector is then passed to a Dense Neural Network (DNN) to predict whether the answer is ”yes” or ”no”, based on the image and question.

The DNN consists of three layers. The layer sizes are 1024, 512, and 1, respectively. The ReLU activation function is applied to every layer of the network, except the output layer, which uses the sigmoid activation function. Dropout (0.4) is applied to the concatenated feature vector, as well as to the output of each of the network's layers to prevent over-fitting.

During the training of the DNN, the weights of both the encoder network and RNN are frozen. The DNN is trained for five epochs, with a learning rate of 2e-4, with a batch size of 128.

Fig. 3: Network structure for the vision model



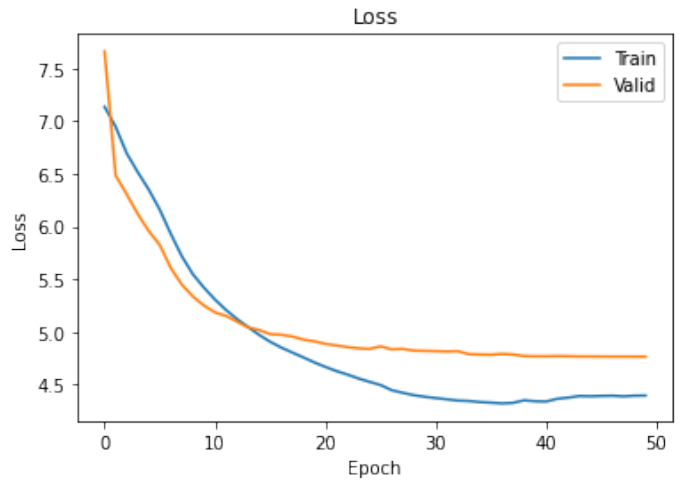


Fig. 4: Graph indicating the change in training and validation loss for the language model over training epochs

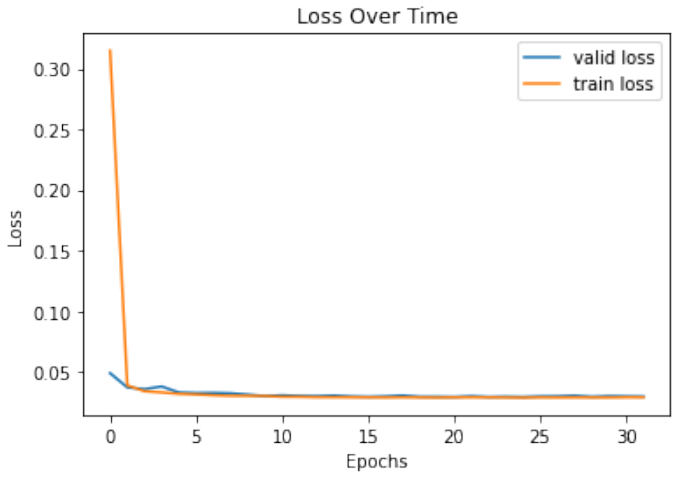


Fig. 5: Graph indicating the change in training and validation loss for the vision model over training epochs

IV. Results

A. Pretrained Models

The RNN had a final train (cross-entropy) loss of 4.5 and a final validation loss of 5.0. Moreover, the model had a train perplexity of 81.164 and a validation perplexity of 117.412.

Fig.4 show the change in training and validation loss over training epochs.

The convolutional autoencoder had a final train loss of 0.029 and a validation loss of 0.030, where the loss is calculated as the mean-squared-error between the original image and the reconstructed image.

Fig. 5 show the change in training and validation loss over training epochs.

B. Modular Network

At the end of training, the model had a training accuracy of 69.9 percent and a validation accuracy of 84 percent. The large increase in accuracy between train and validation sets is likely due to dropout, which is only used while the model is training.

Fig. 6 show the change in training and validation loss over training epochs.

The final model had a test accuracy of 77.8 percent. This accuracy is about 28 percent better than could have been achieved by random chance and 13 percent better than could have been achieved by guessing ”yes” for every question. Thus, the model seems to have successfully made use of the lingual and visual information to answer yes or no questions.

Fig. 7 show the change in training and validation loss over training epochs.

V. Conclusions and Future Work

Visual Question Answering is an exciting machine learning task that has many possible applications. Thus, one can expect much more research in this domain in coming years. While transformers seem to be the most popular current approach, modular networks remain a viable and easy-to-interpret method. The modular network designed in this paper was successful in it’s ability to use both lingual and visual

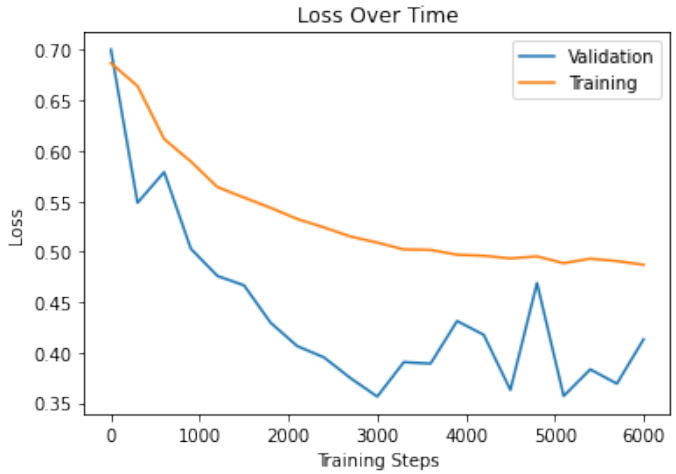


Fig. 6: Graph indicating the change in training and validation loss for the composite model over training epochs

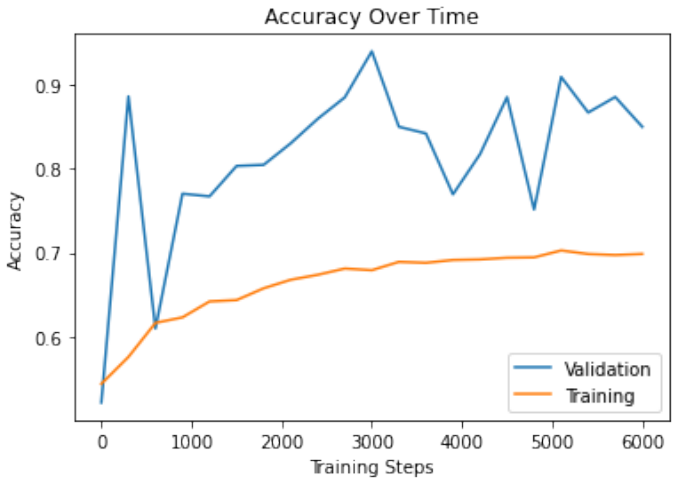


Fig. 7: Graph indicating the change in training and validation accuracy for the composite model over training epochs

information, however, more could have been done. For example, different pretraining methods could have been performed and then tested on the VQA data. For the lingual model, a question-and-answer dataset could could have been used to better prime the modular network to answer questions. For the visual model, many other pretraining tasks could have been tested, such as a Generative-Adversarial Network (GAN). The discriminator portion of the GAN could then be used in place of the encoder network in our current architecture.

VI. Conclusion

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: visual question answering," *CoRR*, vol. abs/1505.00468, 2015. [Online]. Available: http://arxiv.org/abs/1505.00468

[2] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. Van Den Hengel, "Visual question answering: A survey of methods and datasets," *Computer Vision and Image Understanding*, vol. 163, pp. 21–40, 2017.

[3] A. K. Gupta, "Survey of visual question answering: Datasets and techniques," *arXiv preprint arXiv:1705.03865*, 2017.

[4] M. Malinowski and M. Fritz, "A multi-world approach to question answering about real-world scenes based on uncertain input," *Advances in neural information processing systems*, vol. 27, 2014.

[5] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, "Visual7w: Grounded question answering in images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4995–5004.

[6] L. Yu, E. Park, A. C. Berg, and T. L. Berg, "Visual madlibs: Fill in the blank description generation and question answering," in *Proceedings of the ieee international conference on computer vision*, 2015, pp. 2461–2469.

[7] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, "Are you talking to a machine? dataset and methods for multilingual image question," *Advances in neural information processing systems*, vol. 28, 2015.

[8] M. Ren, R. Kiros, and R. Zemel, "Image question answering: A visual semantic embedding model and a new dataset," *Proc. Advances in Neural Inf. Process. Syst*, vol. 1, no. 2, p. 5, 2015.

[9] K. Kafle and C. Kanan, "Answer-type prediction for visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4976–4984.

[10] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1–9.

[11] K. Chen, J. Wang, L.-C. Chen, H. Gao, W. Xu, and R. Nevatia, "Abc-cnn: An attention based convolutional neural network for visual question answering," *arXiv preprint arXiv:1511.05960*, 2015.

[12] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *European conference on computer vision*. Springer, 2016, pp. 451–466.

[13] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," *Advances in neural information processing systems*, vol. 29, 2016.

[14] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 39–48.

[15] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *International conference on machine learning*. PMLR, 2016, pp. 2397–2406.

[16] "Understanding lstm networks," https://colah.github.io/posts/2015-08-Understanding-LSTMs/, accessed: 2010-09-30.

[17] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," 2016.

[18] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll'a r, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[19] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6904–6913.