GRUNet: A Novel Bi-directional RNN for VQA

Vedang Lad
MIT
Cambridge, MA
vedlad@mit.edu

Michael Hensgen
MIT
Somerville, MA
mhensgen@mit.edu

Abstract

With the growing popularity of the Visual Question and Answering (VQA) dataset along with the introduction of various state of the art image and language models, we hope to push the accuracy of the VQA dataset by combining existing image and text encoders, to create a novel model using a method never implemented before. The model is tested on the existing VQA dataset V2 [1]*. Limited by computing power, we combine pre-trained frameworks and fine-tune existing image and text encoders to better optimize them for the VQA dataset. While there was no VQA Challenge this year, we achieved competitive results on the "Abstract Scene" image set using a novel framework. We test variations of ResNet, LSTM, and GRU architectures and then create a novel Modified Bidirectional RNN.*

After rigorous testing which we present in the paper, we propose a new framework which we call GRUNet. GRUNet is a novel Bi-Directional RNN architecture that combines GRU + RNN + ResNet to effectively combine text and image input to answer VQA questions. This architecture represents a popular style of fusion of vision-language models that are known to be effective at VQA.

1. Introduction

1.1. Motivations

Understanding a scene, or the problem of scene segmentation is essential in nearly all forms of Computer Vision (CV). Similarly, parsing text input from humans or other computers is another cornerstone of Natural Language Processing (NLP). Both represent the forefront of research, seen by the various popular models that handle both image and text as inputs such as CLIP [3]. The principles of VQA are based on the fact that in the presence of multimodal knowledge input, here through CV and NLP, an intelligent model can answer correctly regardless of noise or complexity - the way a human would. A task that is simple for humans but not for computers, the success of the VQA dataset represents a step forward towards a compelling "AI-complete" task. An AI-complete task, as discussed in the introduction of the first VQA dataset [1], represents just one piece in the quest for artificial intelligence. Other related tasks include but are not limited to Video Question and Answering, image captioning, and image generation. As pictured in 1, we have an example of an abstract scene with a question, which we tackle in this paper.

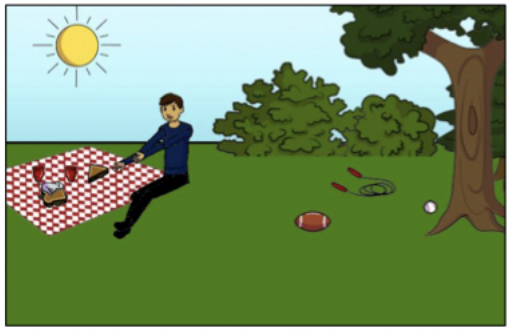

Figure 1. An examples of an abstract scene from [1] which we train and test on in the paper

We choose to create a model that answers open-ended questions as opposed to multiple-choice questions. This represents a more AI-complete task, as there is no answer suggestion. Accuracy is determined simply by

$$Acc(ans) = min(\frac{\text{humans that said } ans}{3}, 1)$$

An exciting application of VQA datasets is building models that can serve as visual aids for the blind. The idea is that a robust model can be pointed at new scenes and answer questions about the scene or subject to a level of accuracy to that of a human.

Since we choose to investigate abstract scenes, we have access to 200,000 questions and just 50,000 images for both testing and training - far less than that of real images. While

[1] (https://visualqa.org)

abstract scenes are "simpler" than real images in the sense that they do not contain adversarial noise or detractors, dealing with far less training and testing data requires models to train efficiently. This is a task that we specifically test before making GRUNet.

In this paper, we test a variety of architectures to determine which performs most optimally on VQA v2 testing datasets. We build on the existing network presented by [1], by modifying the image encoder from VGG18 to various ResNets. We also modify the text encoders using Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM). We combine the principles of a fusion mode from [6] and our best image and text encoders to create GRUNet. All tested models can outperform the [1] baseline which achieves only a 0.55 average score on testing sets. Our main contribution in this work is suggesting a network architecture that has not been used before. Using transformer architectures, as discussed in [6] and [5] has historically not performed highly on VQA data sets following rigorous hyperparameter tuning, which was therefore avoided. We focus here on various RNN structures and propose a new solution.

2. Related Work

2.1. Dataset and Baseline

Aishwarya et al [1] collected the VQA dataset and created baseline architectures that achieve reasonable performance. In the abstract dataset, they create 50,000 abstract scenes, split into 20K train, 10K validation, and 20K test. A scene consists of a question, an image, and an answer or a set of answers. For the Open-Ended task, a model must produce an answer, and that answer is considered correct if it appears in the set of correct answers. Humans, when given a question and image, can produce the correct answer 83% of the time for real images and 87% for abstract images. The current state-of-the-art models, as discussed below, do not achieve that accuracy, showing that there is room for improvement in VQA models.

The baseline architecture created [1] uses a text encoder and an image encoder multiplies the corresponding features and then uses a fully connected layer to decode into words from the vocabulary. The text encoder uses an LSTM with input word2vec embeddings, and the image encoder uses a convolutional layer fed into VGG-18. Their model achieves 57% accuracy on real images and 55% accuracy on abstract images.

2.2. Joint Image and Text Encodings

Successful follow-up work on VQA has used joint encodings of images and text instead of encoding separately. CLIP [3], a joint language-image transformer, was pretrained to maximize the cosine similarity between image and text embeddings. When fine-tuned on VQA, the CLIP-ViL model [5] achieved 76% accuracy on real images.

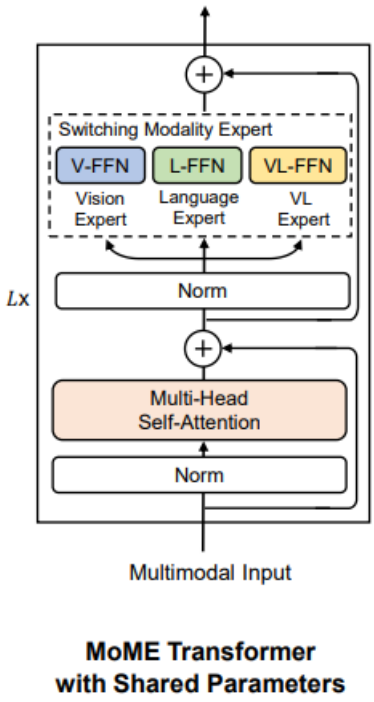

Figure 2. VLMo architecture [6]

VLMo [6] provides another transformer architecture for VQA that uses joint image and text encodings to improve learning. They have three modality expert feedforward neural networks: a vision expert, a language expert, and a vision-language expert. Their architecture is shown in Figure 2, and they pre-train on image-text contrast, image-text matching, and masked language modeling. While we don't use a transformer model, we take direct inspiration from their architecture, especially their image-text matching architecture (See Figure 3). VLMo achieves a similar accuracy to CLIP on real images, at 76%, rounding out the state of the art accuracy.

As discussed by [6], dual image and text encoders oftentimes do not work as well for VQA. As seen by [5], RNN architectures that can sequentially parse a large input work best for text encoders. In a similar vein, image encoders like ResNet and BGG trained on millions of images are hard to compete against.

2.3. Bidirectional Recurrent Neural Networks

Recurrent Neural Networks (RNNs) and their variants are effective at normal question answering tasks. They do so effectively because they maintain the memory of the question with a sequential encoding of the inputs and passing

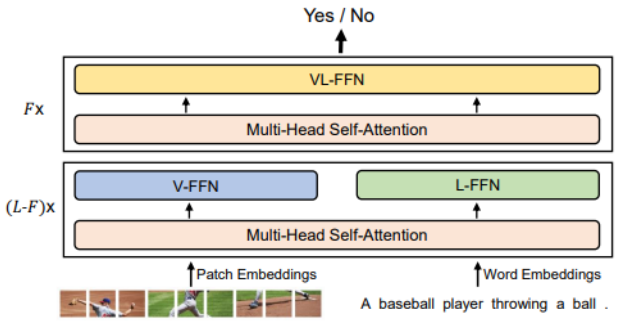

Figure 3. VLMO pretraining for the image-text matching. We base GRUNet on this framework with recurrent memory units on feeding into and out of an encoder

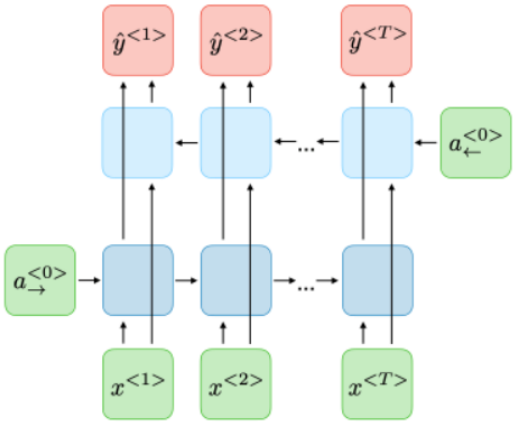

Figure 4. A traditional bidirectional RNN architecture. x is the input sequence, a is an optional input encoding, and y is the output.

that sequential encoding, along with the next input, into the next unit of the network (See Figure 4 for a diagram). A Gated Recurrent Unit (GRU) is set up essentially the same way as an RNN, just with an additional forget gate where it can learn when to forget information. Long Short Term Memory (LSTM) adds a learn gate, a remember gate, and a used gate on top of the forget gate.

Bidirectional RNNs, LSTMs, and GRUs significantly outperform bag of words and deep-question answering baselines in factoid question answering [7]. Bidirectional RNNs learn more about the sentence structure than a unidirectional RNN because words at the end of a sentence can affect the meaning of words at the beginning – the meaning of a sentence flows bidirectionally. We incorporate Bidirectional RNNs, LSTMs, and GRUs for the GRUNet Architecture because they still perform well on question answering without requiring the same pre-training of attention/transformer architectures.

3. Approach

3.1. Text Encoder

We start with the existing architecture which achieves 55% percent accuracy. Our first step is to optimize the existing method and fine-tune it. This is achieved with the addition of linear layers that connect the image and text encoders. We call this "baseline-finetune". This existing architecture utilizes LSTM which we then replace with GRU. An LSTM's ability to keep track of long sequences makes them an ideal candidate for text encoders, which in this case questions to be answered. The idea was to keep the principles of an LSTM but to see if the introduction of the forget-gate in a GRU can help to learn on the VQA dataset. Furthermore, GRU's requirement of fewer parameters was hypothesized that it will allow for faster training and learning. This was essential in training and testing since these are large datasets that can not be run locally but need to be tested on all test images to have a meaningful metric. The results of these investigations can be seen in 4.

3.2. Image Encoder

We experiment and fine-tune the VGG framework and ResNet frameworks. We utilized pre-trained models, which are trained on ImageNet. Here we investigated how the number of layers in architecture, specifically in ResNet affected the image detection in the abstract scenes. While larger ResNets are computationally heavier, we wished to see which architectures would be optimal for GRUNet. These results are more rigorously presented in 4/

3.3. GRUNet Architecture

GRUNet is another architecture that attempts to join the text and image encodings to improve VQA output and combines that idea with the successful memory of bidirectional RNNs. The main idea of GRUNet is that we want each word of the sentence to be paired with an encoding of the image specifically based on that word, and then we encode the joint encodings temporally. This way we have some memory over the entire sentence's image + text encodings.

Figure 5 diagrams the GRUNet Architecture. Specifically, our model first creates a word2vec [2] embedding of the sentence and then feeds those embeddings into a GRU. At each timestep, the GRU outputs an encoding, which we feed into an encoder that uses ResNet on the input image and combines with the text encoding via a fully connected layer. We feed the output of the mid-layer encoder into another GRU for each word. The output of our final GRU layer passes through a fully connected decoder layer to produce answers in the vocabulary space.

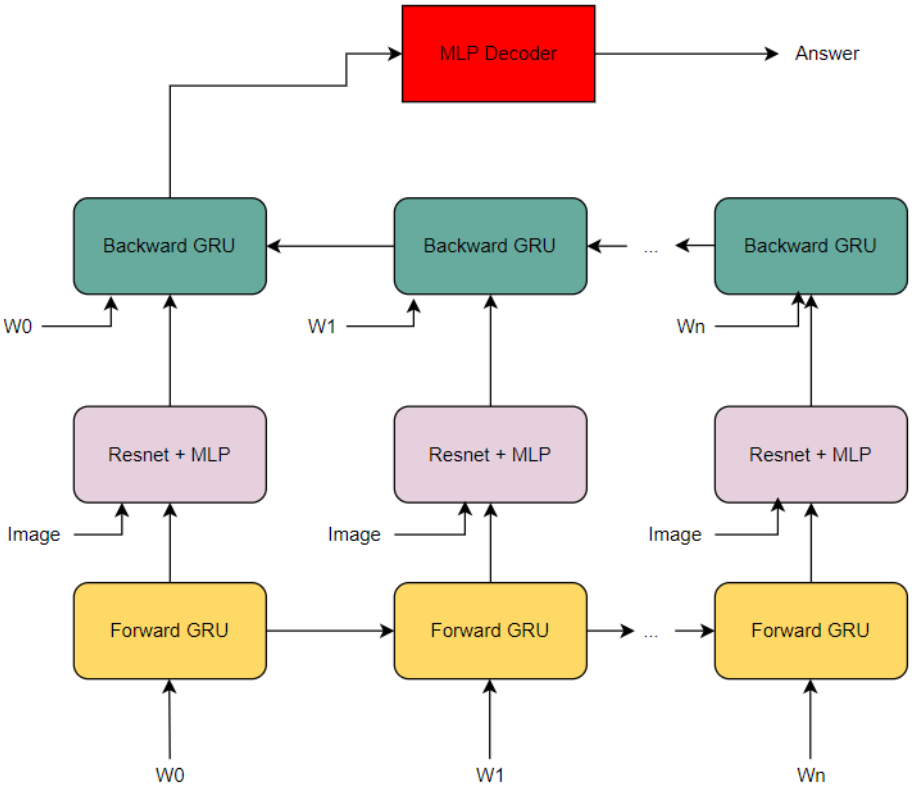

Figure 5. Our GRUNet architecture. For a word of length n, Wn is the embedding of the nth word of the sentence. For the forward GRU layer, we pass in only the word embedding at each timestep. Then, to the ResNet + MLP layer, we input the image and the forward GRU's output (this should produce our word + image encoding), We reinput the word along with the joint encoding into the backward GRU, whose output is decoded by a simple fully connected layer.

4. Results

Following the testing of the various architectures, we find that the best Image Encoder is one that uses ResNet with 152 layers. It is crucial to note that ResNet50, as well as VGG18, achieve similar accuracy, just marginally less. We chose to use the best implementation in GRUNet as running for only 15 epochs allowed the use of the computationally heavier neural network. VGG, known to be faster than ResNet but less accurate performs extremely well and is also a suitable image model for an efficient model however we chose to go with the state of the art on accuracy which was ResNet152.

On the text encoder, we found that the best was, as the title of this paper suggests, the GRU architecture. This model learned faster than any of the other models, achieving close to 0.50 accuracy on the test set just after one epoch. In two epochs, it outperforms the baseline in [1]. The smaller network on its own outperforms the 152 layer ResNet, which justified its use in GRUNet.

While GRUNet does not achieve the state of the art, it provides a unique architecture that is within the variance of other state-of-the-art models. It is a method that has been implemented before and was an interesting method of combining Text and Image information. We attribute the lack of performance to the introduction of a more complex architecture than needed for the abstract dataset. Where a bidirectional RNN here may seem overkill, a possible future work could be to apply these models to the more complex real images dataset. With more images in ImageNet coming from real images as opposed to abstract scenes, we believe this my help performance.

After tuning the parameters of our model (training for 15 epochs with a batch size of 256), our models all outperform the original baseline architecture accuracy of 55% but do not reach the state-of-the-art accuracy achieved by transformers. The accuracy of all the models is shown in 8.

5. Discussion

Overall, we saw improvements over the baseline by changing the encoders and decoders but didn't see any im-

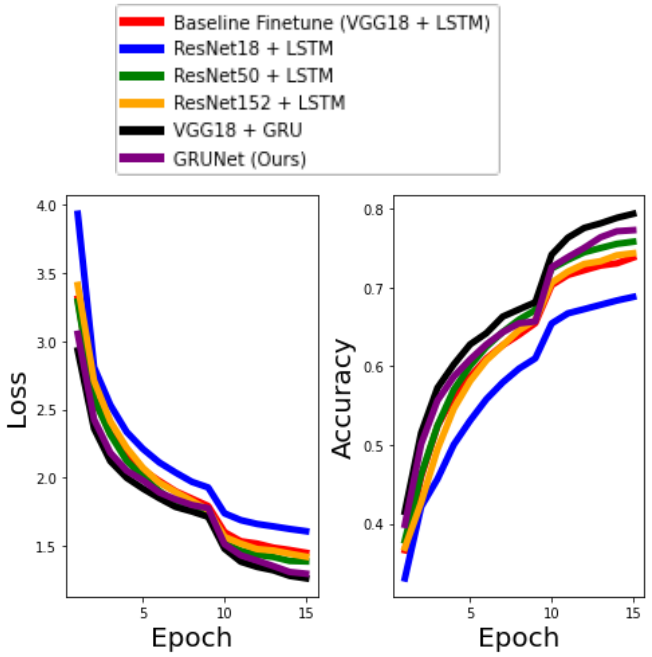

Figure 6. Top: Training Loss and Training Accuracy on VQA

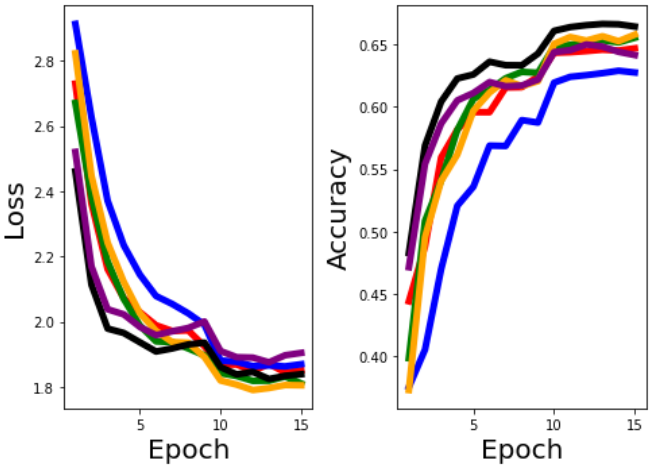

Figure 7. Top: Training Loss and Training Accuracy on VQA As seen by the purple line in the plots, GRUNet does not achieve the state of the art but is competitive with the remaining models. The best model in the architecture is VGG18 and GRU

provement when changing the GRUNet architecture. One reason for this could be that we kept the decoder (just two fully connected layers) consistent between both models. Our idea with that was to be able to evaluate only the encoders and see if our joint encoder would perform better than separate encoders. However, this may not have been the optimal approach because RNNs are structured to produce output words/embeddings at each timestep, so perhaps simply taking the output of the final GRU cell would have been better instead of using the encoding produced by the final GRU layer.

Model	Accuracy
Baseline	54.72
Baseline Finetune	64.67
ResNet18 + LSTM	62.74
ResNet50 + LSTM	65.52
ResNet152 + LSTM	65.77
VGG18 + GRU	66.42
GRUNet (Ours)	**64.22**

Figure 8. Model Validation Accuracy. We we do not achieve the state of the art, we present a new method that is within variance of successful methods.

Another reason for GRUNet's mediocre performance could have been hyperparameters. We observed the hyperparameters working for the separate encoders, but the loss of GRUNet does have multiple timesteps where it's increasing, and it ends at the highest loss while being near the top in accuracy. It makes sense that GRUNet's optimal hyperparameters would be different than the baseline and other encoders because of its large shift in architecture. Perhaps if we lowered the learning rate the loss decrease would be steadier, or if we increased the batch size. We could try a grid search over hyperparameters in the future to see if that's the issue.

One final issue with our architecture is that the joint encoding isn't learned as directly as in transformer models. In models like CLIP and VLMo, they specifically train for image-text contrast and use complex combinations of them, such as cosine similarity between the encodings, to ensure the contrast is learned. In our model, we do no such pretraining and only have a fully connected layer connecting the encodings of the Forward GRU and Resnet as the input to the backward GRU. Therefore, playing around with the Resnet + MLP layer and possibly changing the method of doing the joint encodings could improve the model. Additionally, we could use self-attention before the RNN to improve the word encodings.

If one were to create an architecture that works best, for future works we suggest a combination of GRU with ResNet, without a bidirectional RNN. Since GRU and ResNet both perform the best on their own, we hypothesize based on the tests that this would be the simplest and the best architecture

6. Conclusion

Overall, GRUNet is comparable to alternative models with separated encoders and decoders. Future work could focus on improving the Resnet + MLP layer to a more complex joint representation of image + text timestep. One pos-

sible future model would be to change that middle layer to create a CLIP encoding of the word + image combination for each word in the sequence, and then feed that into a backward or bidirectional RNN to possibly improve on CLIP's already state-of-the-ard VQA performance.

Another new model could be completely retraining RESNET on an abstract image dataset. Since it's currently trained on mostly real images, using only abstract images could improve its performance on the abstract dataset. Another group's project from a couple of years ago used a generative adversarial network to generate new abstract images for the VQA dataset, so these could be used to pre-train a ResNet encoder for abstract VQA.

Other encoders we would like to try in the future include Merlot [8] and DALL-E [4]. Merlot is a transformer pretrained on image/text contrasting in youtube videos along with temporal ordering of the frames. It performs very well on video question answering, so it would be cool to see how that performance would translate to using it out of the box on VQA, fine-tuning it, and substituting it into the GRUNet middle layer. DALL-E can create images based only on a text description, so that could perhaps be used to encode the image + text based on the real image's reasonability/proximity based on a set of DALL-E created images for that text.

There are many more ways to encode and pre-trained encoders to explore, and we hope they will push the boundaries of Visual Question Answering accuracy higher.

7. Individual Contributions

7.1. Vedang Lad

In this final project, my contribution came from making around half of the models that are used, including ResNet, GRU, and RNN. Having a Colab Pro account from another class, I was responsible largely for running all of the models at hand and debugging the integration of various architecture. I also constructed the various training and testing plots seen in the report and presentation. I contributed a significant portion of both the status update report as well as the final report and presentations. A unique individual responsibility I had was creating a schedule with my partner to make sure that we get the project done on time.

7.2. Michael Hensgen

My largest responsibility for this project was coding up and debugging the GRUNet model since while Vedang set up the training on his Colab Pro I could debug the models we were making on my basic Colab account. I would check that all the outputs were what we expected so that the long training processes would be correct. In the final report, I wrote the related work and collaborated on the approach, results, discussion, and conclusions sections of the final report. I also tried some other experimental models that didn't work out as well (and we couldn't get computationally time efficient enough), such as an architecture with a BERT encoder.

8. Reproducible

All of the code required to replicate the project can in be found in the public repository. https://github.com/vdlad/basic$_vqa$

References

[1] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. Vqa: Visual question answering, 2015. 1, 2, 4

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. 3

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. 1, 2

[4] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021. 6

[5] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks?, 2021. 2

[6] Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts, 2021. 2

[7] Dong Xu and Wu-Jun Li. Full-time supervision based bidirectional rnn for factoid question answering, 2016. 3

[8] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models, 2021. 6