Attention on Attention: Architectures for Visual Question Answering (VQA)

Alex Nutkiewicz, Jasdeep Singh, Vincent Ying

{alexer, jasdeep, vhying}@stanford.edu

OVERVIEW

Visual Question Answering (VQA) is an increasingly popular topic in deep learning research as it requires coordination of several artificial intelligence-related disciplines, including Computer Vision and Natural Language Processing. The inputs to a VQA model are images and free-form, open-ended natural language questions about the image, and the model's goal is to produce a natural language answer about the input.

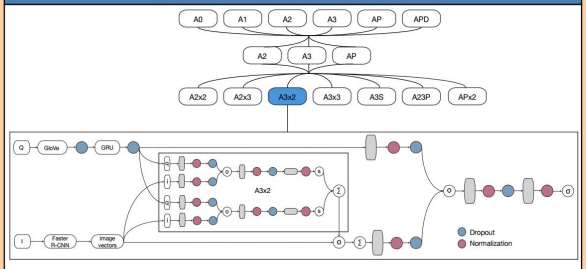
Our project aims to investigate previous methods of implementing VQA and better understand the characteristics of more successful network architectures for this task. We build upon previous iterations of winning VQA Challenge models [1] by developing thirteen attention mechanisms and introducing a simplified classifier to the model. We evaluate our model against other VQA implementations via an evaluation metric used in the VQA Challenge. With a validation set accuracy of 64.78%, we show our model is able to beat the score of the winners from the 2017 Challenge.

DATASETS

While many large-scale datasets have been developed for the application of VQA, we used the VQA 2.0 dataset, which contains over 200,000 images and at least three questions per image. We do the following pre-processing:

- Training questions and answers are tokenized. These tokens are represented using Wikipedia+Gigaword GloVe word embeddings [2]
- Image features are created by passing the images through a Faster R-CNN, which detects object centric elementsin the VQA model [3]

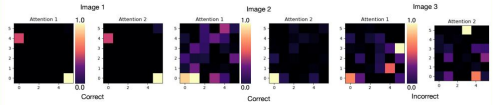
HYPERPARAMETER SEARCH



We learned that one of the biggest determinants for model accuracy were improved attention mechanisms. We experimented with 13 different modules and found that the A3x2 module performed the best. We then performed over 300 hours of hyperparameter tests, looking at each hyperparameter individually. At each step in the search, as shown in the figure below, we found the optimal value and did all subsequent searches using that value. Hyperparameters were determined using validation set accuracy.

RESULTS, DISCUSSION, & FUTURE WORK

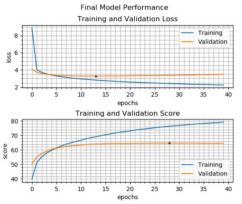
After determining the optimal model through experimentation and tuning, we were able to achieve an evaluation score of **64.78%**, out performing the existing state-of-the-art single model's validation score of 63.15% by [3].



We believe one of the most significant reasons our score was able to beat the state-of-the art results was because of the more sophisticated attention mechanism. The three heatmaps shown above demonstrate how adding a second attention mechanism allows the model to learn different aspects of an input image.

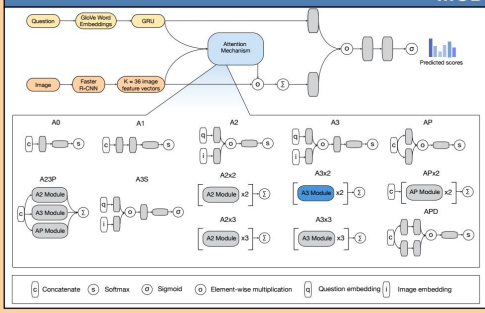
- ***Image 1***: for simple attention tasks both attention mechanisms are able to find the appropriate locations in the image.
- ***Image 2:*** when the task requires focus on multiple locations in an image our model can still predict with high accuracy
- ***Image 3:*** for complicated tasks, the dual attention is confused

Given enough epochs, we found that our model overfit the training set. When compared to other papers, we found this to be expected because there is a large disparity between the distribution of questions in the validation and training set. This is understandable because VQA is an open ended task.



While our computational and time resources were limited, we were able to begin an extensive architecture and hyperparameter search. Our future work would look at the synergistic effects of some of these hyper-parameters, as well as experiment with how a bi-directional attention mechanism would impact performance.

MODEL ARCHITECTURE



Preprocessed question inputs are passed through a GRU to create a question embedding. They then join the preprocessed image feature vectors and are passed through a dual one-way, top-down attention module (we use the A3x2 attention mechanism in our final model).This module computes the relevance of each of the image vectors to the current question embedding:

$$a_i = w f_c(f_a(\hat{i}) \circ f_b(\hat{q}))$$

$$a'_i = w' f_{c'}(f_{a'}(\hat{i}) \circ f_{b'}(\hat{q}))$$

The attention weights are normalized using a softmax function. The final image embedding is created by taking a weighted sum of the original image vectors and using the attention scalars as weights.

$$\hat{v} = \sum_{i=1}^{K} (\alpha_i v_i)$$

The final image vector and question embedding are modified in their dimensions, element-wise multiplied, and the resulting joint embedding is then passed through a simple classifier. We use binary cross entropy to calculate the loss during training.

$$accuracy = \frac{1}{m_v} \sum^{Val} (one\ hot(argmax(\hat{y})) \cdot y) \qquad \mathcal{L} = \frac{1}{m_t} \sum_{i=1}^{m_t} -(y \log(\hat{y}) - (1-y) \log(1-\hat{y}))$$

REFERENCES

[1] Teney, Damien, Peter Anderson, Xiaodong He, and Anton van den Hengel. "Tips and Tricks for Visual Question Answering: Learnings from the 2017 Challenge." arXiv preprint arXiv:1708.02711. 2017.
[2] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.
[3] Anderson, Peter, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. "Bottom-up and top-down attention for image captioning and VQA." arXiv preprint arXiv:1707.07998 (2017).