Visual Question Answering with Deep Learning

This blog contains the implementation of “Hierarchical Question-Image Co-Attention for Visual Question Answering” paper in Keras/Tensorflow.

Tulrose Deori Jun 2, 2020 · 6 min read ★

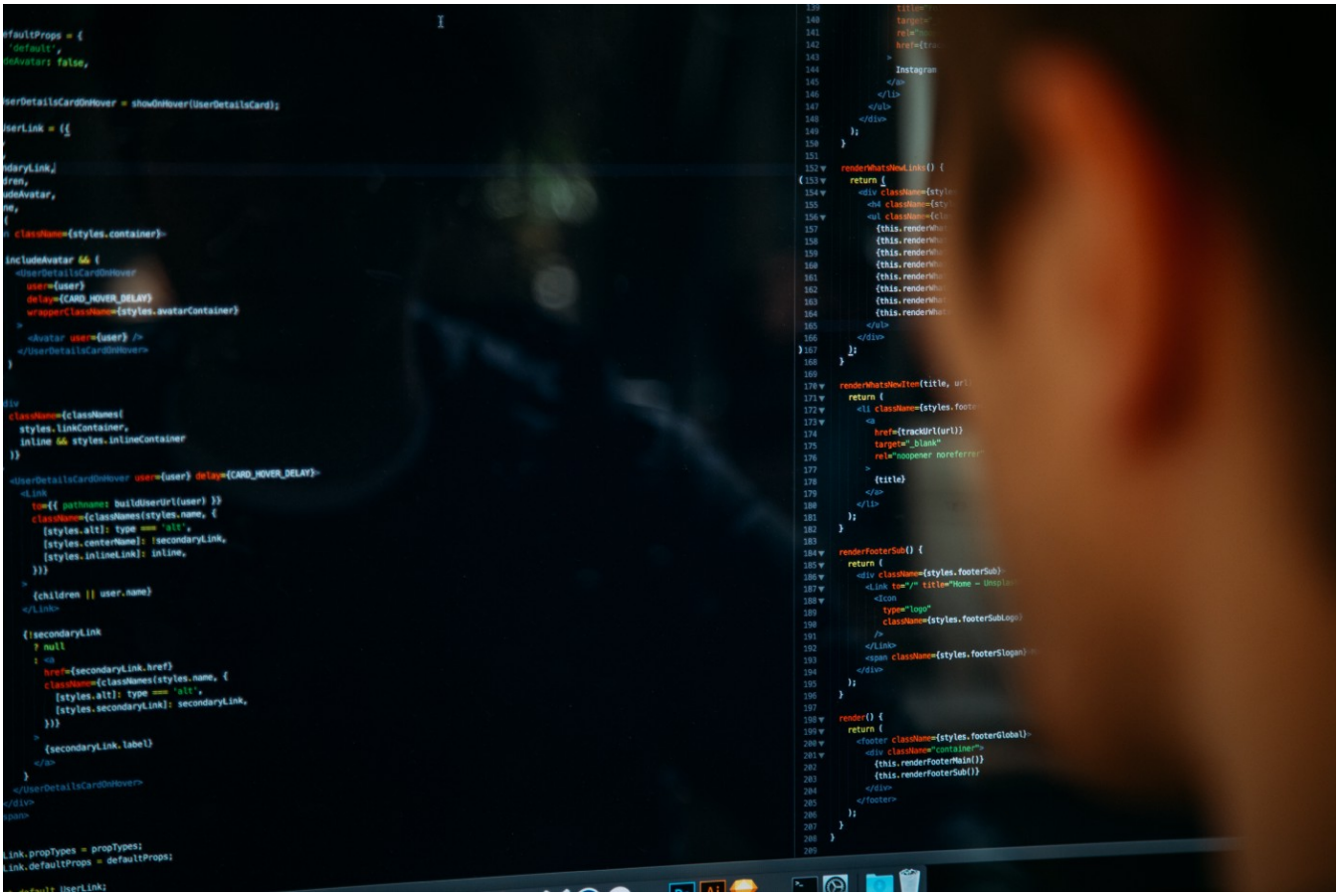

Photo by Charles Deluvio on Unsplash

Table of Contents:

1. Introduction:

Visual Question Answering is a research area about building an AI system to answer questions presented in a natural language about an image.

A system that solves this task demonstrates a more general understanding of images: it must be able to answer completely different questions about an image, oftentimes even addressing different sections of the image.

Let's look at a few examples:

Source: Original VQA Paper

For all the images, our AI system should be able to localize the subject being referenced to in the question and detect it and should have some common-sense knowledge to answer it.

For instance, in the first image, and for the question "*What is the mustache made of ?*": our AI system should be able to figure out that the subject being referenced to is the women's face, more specifically the region around her lips, and should be able to detect the banana.

2. Business Problem:

As humans, it is easy for us to see an image and answer any question about it using our commonsense knowledge. However, there are also scenarios, for instance, a visually-impaired user or an intelligence analyst, where they want to actively elicit visual information given an image.

answer as the output.

The system will answer a question similar to humans in the following aspects:

1. it will learn the visual and textual knowledge from the inputs (image and question respectively)
2. combine the two data streams
3. use this advanced knowledge to generate the answer

3. Understanding the data:

VQA is a dataset containing open-ended questions about images. These questions require an understanding of vision, language and commonsense knowledge to answer.

- 82,783 images (COCO train images)
- At least 3 questions (5.4 questions on average) per image (443,757 questions)
- 10 ground-truth answers per question (443,7570 answers) from unique workers

4. Mapping the real-world problem to an ML/DL problem:

- **Type of Machine Learning Problem:** We pose the problem at hand as a K-class classification problem; where K is the number of a fixed set of answer types in the dataset.
- **Performance Metric:** We evaluate our AI system by the number of questions it answers correctly. The following accuracy metric is used:

$$accuracy = min(\frac{\#\text{ humans that provided the predicted answer}}{3}, 1)$$

5. Understanding the model:

Let's try to understand the method proposed in the paper "Hierarchical Question-Image Co-Attention for Visual Question Answering".

Paper Overview:

source: https://arxiv.org/pdf/1606.00061

All the papers on VQA before this, focused mainly on visual attention. This paper proposes to focus on question attention too. Specifically, this paper presents a novel multi-modal attention model for VQA with the following two unique features:

1. **Co-Attention:** This paper proposes a novel mechanism that jointly reasons for visual attention and question attention, which is referred to as **co-attention**. More specifically, the image representation is used to guide the question attention and the question representation(s) are used to guide image attention.

2. **Question Hierarchy:** Builds a hierarchical architecture that co-attends to the image and questions at three levels: (a) word-level, (b) phrase level, and (c

a) **At the word level**, the words are embedded in a vector space through an embedding matrix.

b) **At the phrase level**, 1-dimensional convolution neural networks are used on the word representations with temporal filters of varying support, to capture the information contained in unigrams, bigrams, and trigrams, and then combine the various n-gram responses by pooling them into a single phrase-level representation.

c) **At the question level**, a recurrent neural network is used to encode the entire question.

For each level of the question representation in this hierarchy, joint question and image co-attention maps are constructed, which are then combined recursively to ultimately predict a distribution over the answers.

Method:

The paper proposes two co-attention mechanisms that differ in the order in which image and question attention maps are generated. The first mechanism, which is called **parallel co-attention**, it generates image and question attention simultaneously. The second mechanism is called **alternating co-attention**, it sequentially alternates between generating image and question attentions. These co-attention mechanisms are executed at all three levels of the question hierarchy.

In this blog, we will discuss the implementation of the Parallel Co-Attention model.

Parallel Co-Attention:

source: https://arxiv.org/pdf/1606.00061

Parallel co-attention attends to the image and question simultaneously. The image and question are connected by calculating the similarity between image and question features at all pairs of image-locations and question-locations. Specifically, given an image feature map $V \in R^{(d \times N)}$, and the question representation $Q \in R^{(d \times T)}$, we calculate something called affinity matrix $C \in R^{(T \times N)}$ as follows:

Considering this affinity matrix C as a feature, image and question attention maps are predicted in the following way:

Based on the above attention weights, the image and question attention vectors are calculated as the weighted sum of the image features and question features, i.e.,

The parallel co-attention is done at each level in the hierarchy, leading to $\mathbf{v}^r$ and $\mathbf{q}^r$ where $r \in \{w,p,s\}$.

source: https://arxiv.org/pdf/1606.00061

6. Implementation Details:

- We don't directly use the image as input into the model. The image is scaled to 224× 224, and then the activations from the last CONV layer of VGGNet19 are

- We use the KERAS tokenizer to extract question features.
- The final input features are: a) image features of shape [49, 512] and b) question features of shape [22,], where 22 is the sequence length of the questions after pre-processing.
- We use the ADAM optimizer with a base learning rate of 1e-4.
- We set the batch size to be 300 and train for 60 epochs.
- We use regularization in every layer.

7. Code:

Architecture:

Custom Layers:

Some custom layers are used to define the model above. Below is the code for the custom layers:

8. Results:

The accuracy of the paper on the VQA 2.0 dataset is 54%. My implementation accuracy is 49.28%.

Below is the demo video of the working model:

same accuracy as the paper, we can try to train the model for larger epochs, and with learning rate decay.

Scaling the images to 448× 448, and then extracting the [512 x 14 x14] activations from the last CONV layer of VGGNet19 to use as the image features can also lead to an increase in performance.

And that's all. Thank you for reading my blog. Please leave comments, feedback, and suggestions if you feel any.

Full code on my GitHub repo, *here*.

You can find me on LinkedIn, *here*.

10. References:

- Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh, Hierarchical Question-Image Co-Attention for Visual Question Answering (2016)
- https://github.com/ritvikshrivastava/ADL_VQA_Tensorflow2
- https://www.appliedaicourse.com/

Back to top ^