Lightweight VQA

Nabih Nebbache Zeryab Moussaoui Leila Hamdad

1 Introduction

The Visual Question Answering (VQA) is an AI system that can answer questions about visual content. VQA is still a very new research domain which is mostly based on deep learning, the most successful VQA systems implementations like [1] [2] are based on very big neural networks, and this to increase the result accuracy, this has also the drawback to produce a very large VQA system with slow responding time. This approach can be explained by the research's goal which is to produce the most efficient system in term of accuracy omitting other crucial industrial aspects like speed and memory saving. This document describes our attempt to produce a more lightweight deep learning based VQA architecture compared to state of the art VQA models. This of course is a trade off between accuracy and speed/lightness.

2 Data pretreatment

Before training the model, we must in a first step clean the data that will be fed to this latter. We chose in our work the VQA V2 dataset [3] as a training and testing dataset. VQA V2 elements are composed of an image, a question and a list of answers, hence the pretreatment of that dataset is the pretreatment of each kind of data composing it.

2.1 Image pretreatment

The image pretreatment's goal is to obtain at the end a dense vector representing the image's characteristics. We pretreat the image by applying the following steps:

(i) Image re-scaling in order to homogenize the treatment over the entire dataset

(ii) Characteristics extractions by applying a pretrained ResNet50 on the image and removing the last layer (because it is a classification layer). The ResNet50 network is pretrained on ImageNet [4].

2.2 Question pretreatment

Since deep learning systems accept numeric-only inputs, the question which is composed of strings must be transformed into numeric format. The question's pretreatment passes through the following steps :

(i) Tokenising and removing non-character symbols

(ii) Removing words that have an occurrence inferior to a threshold

(iii) Fixing the length of all questions in order to homogenize the treatment across all the dataset

(iv) Assigning to each word a natural number index

Each question becomes at the end a list of indexes, each index representing a word.

2.3 Answer pretreatment

As for inputs, outputs must be pretreated. We describe in this subsection the steps followed to transform the answers into integers.

(i) Cleaning answers by transforming all words into lower case and removing comas and interrogation marks and full stops.

(ii) Applying a special VQA V2 answers treatment by analysing the answers and replacing words that were misspelled or that have synonyms in the answer corpus with the right words. Table 1 shows some examples of treatments applied to answers.

(iii) Removing answers that have an occurrence inferior to a threshold. This step is very important because besides removing insignificant words, the original output dimensionality is too big to be put into a training model.

Table 1: Table describing VQA V2 pretreatment.

Question type	Original answer	Answer after treatment
Any	No 1	No one
Any	No clue, I don't know, Not sure	No idea
Counting	Letter written numerals	Arabic numerals
Yes/No	Contains "no"	No

(iv) The last step is to transform each question's set of answers into a numeric format. We first assign to each answer an index. Then for each question, the set of its answers is represented by a vector with each its entries representing the score of the answer that have the entry's index for that question. The score is calculated with the equation 1 which is a formula used to process the score in the VQA dataset.

2.4 Dataset creation and serialisation

The last pretreatment step is to create and serialise the dataset. The creation of the dataset is done by regrouping similar data elements together. VQA V2 dataset provides a way to find elements that have similar questions with different images and answers. These similar elements are put one next to the other in the dataset. After forming the dataset, We serialise the data into binary files, the data is then aggregated and the reading is faster.

$$Accuracy = min(Answer_occurrence/3, 1) \quad (1)$$

3 Model Architecture

We will in this section describe the model architecture used. The model is composed of the following modules.

3.1 Question's characteristics extractor

The goal of this module is to extract questions' characteristics. The questions are after pretreatment in the form of an index vector, the final result must be a float vector. To reach this goal, we passed by the following steps:

(i) Embedding each word in an embedding matrix initialised with GloVe [5].

(ii) Feeding each question to a recurrent neural network of type GRU [6]

(iii) Feeding the result of the GRU to a dense neural network with ReLU as activation function

3.2 Image's characteristics extractor

Since a lot of work have been done on the image in the pretreatment phase. The image's characteristics extractor only applies a L2 normalisation on the last layer of the image characteristics.

3.3 Attention system

The goal attention system is to focus on the most important regions of the image, and that by assigning to each of the region of the image a score between 0 and 1. The bigger the score is, the more important the region is. The Attention system is composed of the following steps :

3.3.1 Characteristics transformation

The first step is to transform the image's characteristics to match the last dimension of the question vector. In parallel we apply a broadcasting to the question vector to match the first dimensions of the image vector. This transformation are only done inside the attention system module.

3.3.2 Characteristics fusion

The next step is to fuse the image and question vectors by applying a Hadamard product. The result vector represents a common representation of the image and the question with each region of it representing a region from the image parameterized by the question.

Table 2: Comparaison between our model and Pythia [1]

	Our model	Pythia
Speed of inference (seconds)	0.5	1.6
Number of parameters	54 139 309	130 330 044
accuracy	48.7%	70.01%

3.3.3 Attention vector generation

After obtaining a common representation of the image and the question, we feed each of its region to a neural network with an output layer of dimension 1, the total output is then the number of the image regions. We apply then a softmax function to that output to obtain for each region of the image a score between 0 and 1.

3.3.4 Apply attention to the image

After obtaining the attention vector, we broadcast it to match the image's dimensions, we apply then a Hadamard product between it and the image. The result is a weighted image characteristics. We then sum the result vector on the two first dimensions to obtain a one dimension vector at the end.

3.4 Characteristics fusion

After applying the attention system on the image and obtaining the new image characteristics, we fuse the image and the question by a Hadamrd product to obtain a common representation. Before, we must match the two dimensions, so we apply a non linear transformation on the image via a dense neural network.

3.5 Answer generation

The final phase is to generation an answer. This is done by feeding the common representation generated from the last phase to a neural network whose output dimension is equal to the number of answers in the dataset. Then a sigmoid function is applied to each entry of the result vector to assign a score between 0 and 1 to each answer.

4 Results

We evaluated our model on three aspects: the size of the model, the speed of inference and the accuracy. We compared our model with the Pythia model which was the winner of the VQA challenge 2018 in the table 2. The test where done on a Google Colab machine.

The table 2 shows that our model is more lightweight and more fast than the state of the art model, but it is also less accurate. Of course, the goal of this model is to trade accuracy for speed and lightness, but the accuracy is low mostly because the model hasn't finish training due to resources lack.

References

[1] Yu Jiang et al. "Pythia v0. 1: the winning entry to the vqa challenge 2018". In: *arXiv preprint arXiv:1807.09956* (2018).

[2] Peter Anderson et al. "Bottom-up and top-down attention for image captioning and visual question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6077–6086.

[3] Yash Goyal et al. "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.

[4] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[5] Jeffrey Pennington, 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)*. 2014, pp. 1532–1543.

[6] Kyunghyun Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).