MLDM2 : Visual Question Answering (VQAv2) task

Cedric Osei-Akoto*
oseiaced@students.zhaw.ch
ZHAW School of Engineering

Sithursan Sivasubramaniam*
sivassit@students.zhaw.ch
ZHAW School of Engineering

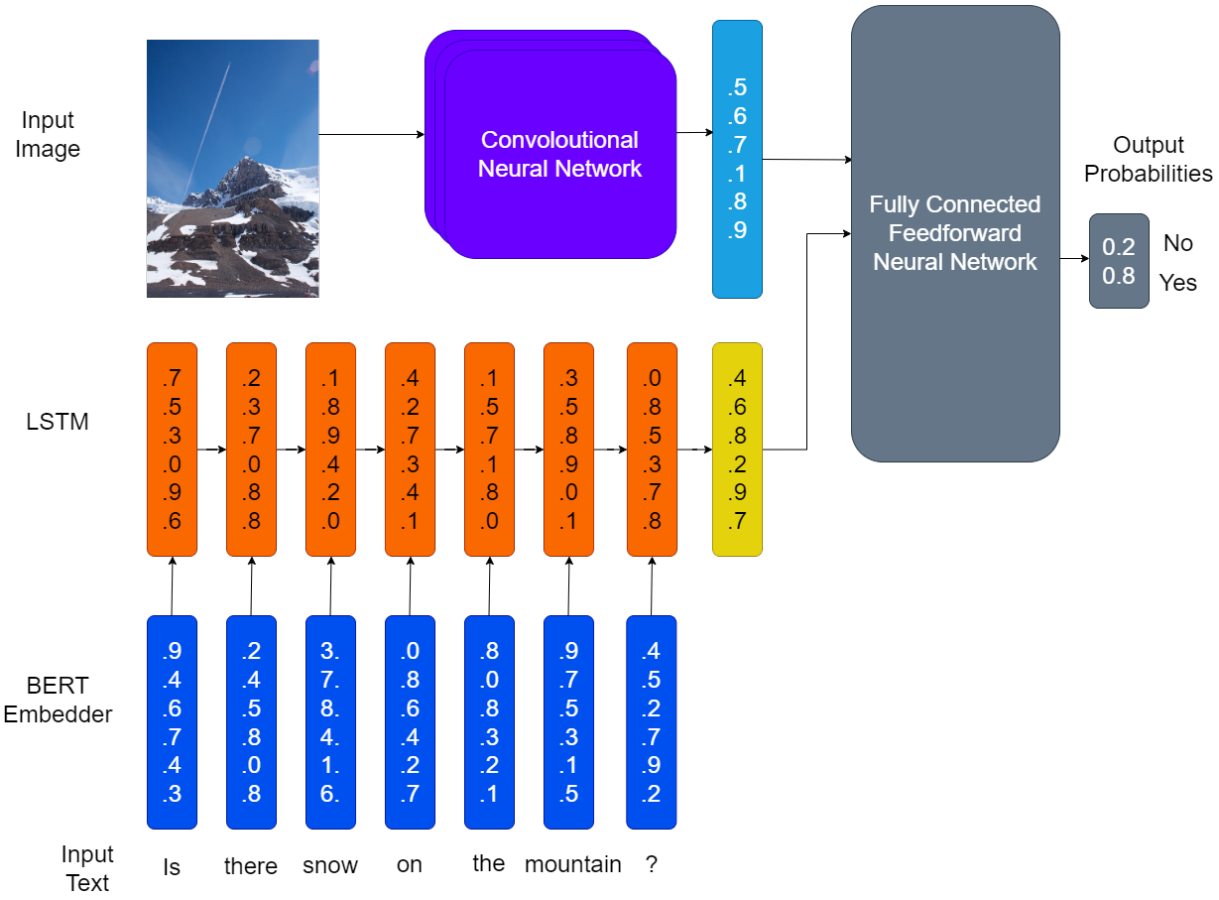

Figure 1: Design of the ML approach

ABSTRACT

In this paper, we present the detailed architecture and preprocessing steps of our CNN+RNN model for the VQAv2 task. We conducted extensive experiments and evaluations to assess the performance of our model and compare it with existing approaches. The results of our experiments demonstrate the effectiveness of our proposed model in addressing the challenges posed by VQAv2 and highlight its potential for achieving accurate and insightful question answering capabilities.

KEYWORDS

datasets, CNN, RNN, VQAv2,

1 INTRODUCTION

As humans we are often confronted with questions about visual circumstances in situations we experience in the real world. Usually such questions urge us to analyse both visual and textual queues in order to answer such a questions appropriately. A key factor in answering such questions manually is the application of textual understanding on to the visual input in relation to the question asked.

With the many technical advancements in machine learning, the interest in answering such kind of questions with computations has risen to a large extent. Such that visual question answering has manifested into it's own domain. Other than in traditional machine learning tasks, VQA (visual question answering) requires multi-modal machine learning models to deduce an answer. Reason for that is the textual and visual inputs, which both are inserted individually to the model, in order to compute an answer leveraging both text and image data. In line with progress in research fields such as natural language processing and computer vision, came first progresses in the VQA domain. Progress, which also proved

that current state of the art multi-modal models solving the VQA task indeed mainly learnt to predict correct answers based on statistical probabilities of textual queues provided. This very issue was addressed with the VQA version 2 task, where the data provided was adjusted to exploit such vulnerabilities in VQA models, leading to a drastic drop in most model performances.
This strong shift in paradigm in the VQA models led us to the question whether an application of textual understanding on a visual input could be replicated and result an unusual accuracy.
In this paper we created a VQA multi-modal model with state-of-the-art uni-modal models as components, which should replicate both textual and image understanding. The output of these components were then concatenated and used as input for a Fully connected Feedforward neural network, which should duplicate the combined understanding.

2 RESEARCH QUESTION

In this research we will mainly focus on concatenating outputs of state-of-the-art uni-modal models and using the resulting data as input for a Fully connected Feedforward Neural Network. Center of attention lies primarily on the accuracy of such a VQA model. Specifically whether an accuracy above 50% can be reached, which would imply that the system can learn relations between visual and textual input, in order to answer VQA tasks.

3 SCOPE OF THE PROJECT

In order to limit the number of potential justifications for research success or failure, only binary VQA tasks were considered for the study.

4 DATASET

Based on our research's urgency to exploit our models functionality in the VQA problem space, the current VQAv2 Dataset was used. This comprehensive Dataset contains a wide range of images linked with corresponding questions for visual question answering (VQA). The images provided in the dataset were initially sourced from the Microsoft Common Objects in Context (COCO) dataset, which entail various scenes, objects, and activities. Each image is paired with multiple-choice questions that cover different aspects such as object recognition, scene understanding, counting, and reasoning. This diversity in data provides a rich and complex visual context for the associated questions. In addition to the questions, the dataset contains metadata describing the question type and answer options,confidence levels and IDs.

5 PREPROCESSING

During the first preprocessing step the VQAv2 dataset was directly loaded from the HuggingFace library in to the code environment. In the next step 20'000 data points were extracted for training of the model, leaving each validation and testing 10'000 data points for the respective tasks.

5.0.1 Preprocessing textual data. In order to adequately preprocess the textual data, a state-of-the-art pre-trained BERT model lowercased, tokenized embedded the input texts. This step was crucial to ensure the controlled handling of varying text lengths and enabled efficient batch processing during model training. The resulting transformed questions were then used as input to the subsequent model for further processing.

5.1 Preprocessing Image data

In terms of image data, we performed resizing and normalization operations to ensure consistency and optimal performance during model training. During the data processing stage, we encountered instances where images referenced Huggingface library dataframe, were not present in our local folder(containing images data directly from the VQA page). To maintain data integrity, we removed these images from the dataframe. The entire preprocessing was executed in batches, enabling us to process data in an adequate manner. Finally, the processed batches were merged to obtain the final dataset for subsequent analysis and model training.

6 MODEL

Our model is a combination of CNN and an RNN network. The aim was to get the strength of Convolutional Neural Networks (CNNs) to capture effective visual information and make use of the Long Short term Memory (LSTM) recurrent neural network to capture sequential textual meaning of the questions. The CNN is implemented as a linear layer with the ReLU activation function. It extracts the important representations of the input images.

Table 1: Model Network Parameters

Layer	Number of Parameters
Image Model (Linear)	4096×1024
Language Model (LSTM)	$300 \times 4 \times 512 + 512 \times 4 \times 512$
Fully connected (Linear)	$(512 + 1024) \times 1024$
Fully connected (Linear)	1024×1024
Fully connected (Linear)	1024×1000

Based on our requirements, we will be utilizing LSTMs due to their ability to overcome certain challenges and potentially provide improved performance. To represent the question, we feed the LSTM with word vectors associated with each token in a sequential manner. The output gate of the LSTM is then used to obtain the representation of the entire question after all tokens have been processed. This vector is of a fixed length and is combined with the 4096 dimensional CNN vector of the image. The concatenated vector is then input to a multi-layer perceptron with fully connected layers. The final layer is a softmax layer that generates a probability distribution over the potential outputs. As Loss Function we've used 'categorical crossentropy'.

6.1 Hyperparameter tuning

We employed Bayesian optimization to automatically search for the optimal hyperparameters for our model. The hyperparameters, including the number of hidden units in the Fully connected Layer and LSTM, image and word vector dimensions, number of classes, batch size, learning rate, and regularization strength, were optimized using Bayesian optimization. The objective function, which was defined as the cross-entropy loss, was minimized during the

optimization process. We used the Adam optimizer to perform the parameter updates. The results were used to train the model 2.

Table 2: Model Hyperparameters

Hyperparameter	Value
Batch Size	32
Learning Rate	0.001
Regularization	0.01
Number of Hidden Units (MLP)	1024
Number of Hidden Units (LSTM)	512
Image Dimension	4096
Word Vector Dimension	300
Number of Classes	1000

7 RESULTS

After training the model and fine-tuning the hyperparameters using Bayesian optimization, we achieved a validation accuracy of 44%. This result indicates that our model effectively learned the underlying patterns and relationships within the training data, leading to accurate predictions for binary yes/no questions in the validation set.

After optimizing the hyperparameters of the model, we further partitioned 5'000 data points from the validation set for Bayesian optimization, while the retaining 5'000 data points for the evaluation. Subsequently, we evaluated the performance of our optimized model on the separate test set comprising 10'000 new data entries. The model demonstrated a test accuracy of 38%, showcasing its ability to generalize and provide accurate predictions on new instances. It should be noted that the model then encountered challenges with questions that involved fine-grained details, ambiguous context, or complex reasoning, leading to lower accuracy scores in those cases. All of the results imply that the system cannot learn relations between visual and textual input, in order to answer VQA tasks, hence needs to be futher audited and adjusted.

8 FURTHER EXPLORATION AND FUTURE WORK

In order to get a deeper understanding of multi-modal VQA models, state of the art implimentations could used to further research the initial question. In addition we could use the full dataset to get a better result.