Visual Question Answering using CLIP model

Ahmad Hazem:7193, Zeyad Zakaria:6764, Nader Mohammed:7115

Pattern Recogonition Assignment 4

1 Introduction

CLIP (Contrasting Language-Image Pre-training) is a neural network model developed by OpenAI that can connect images and text. It is trained on a massive data set of image-text pairs, and can be used to perform a variety of tasks, such as:Image captioning: Given an image, CLIP can generate a text caption that describes the image.Image retrieval: Given a text query, CLIP can retrieve images that are relevant to the query. Zero-shot learning: CLIP can be used to classify images into categories that it has never seen before, simply by providing the names of the categories. CLIP is a powerful tool for connecting images and text, and has a wide range of potential applications. For example, it could be used to improve the accuracy of image search engines, or to create new kinds of interactive visual experiences.

2 Methodolgy

We followed the same steps that was implemented in paper Less Is More: Linear Layers on CLIP Features as Powerful VizWiz Mode.

- **Data Preprocessing:** The vizwiz data set was read using pandas to observe and see how the data set was divided. Afterwards, we followed the paper steps and began to extract true answers from answers column by getting the most common answer in each label, and we used Levenshetien algorithm algorithm as tie breaking in case several labels had the same number of votes, as a result we got a total length of vocab of **5419**.We **ViT-L/14@336px** model due to its reduced size of encoded features of 768. We read the images from their paths after splitting on stratify, preprocess the images, and encoded, then it was saved in a pickle file to reduce time of reading every time the session is restarted. The stated was repeated on the questions, and was concatenated with each other.

- **Model Designing and Implementation:**. The input of the data loader of the model was the encoded image-question , true answer types, true answers , and true answer-ability.The Model was divided into 4 gates: **Main Gate** The encoded image-question was inputted through a normalization layer of a drop out **0.5**, then a fully connected layer of 512 output channel. The Main gate output was branched into 3 branches. The first branch is the **Answer gate** and it passes through a fully connected layer with normalization and dropout layer.

Next is the **Answer Type gate**, which is the third branch, it passes through a fully connected layer then projected on another fully connected layer with the same size of our vocab then it passes a sigmoid layer to be multiplied with the output of the answer gate.The final branch is the **Answer-ability** gate which passes through a fully connected then a sigmoid activation function then a softmax to determine the answer-ability.

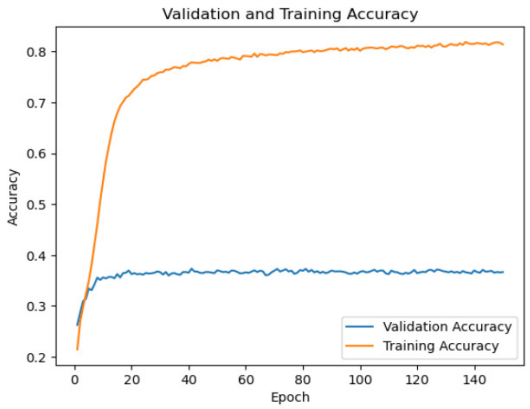
- **Model Evaluation:** We encoded our true answers using one hot encoding, and mapped the answer types to a numeric list , and answerability was just inserted with no changes. We ran our model for 150 epochs, and yielded the following.

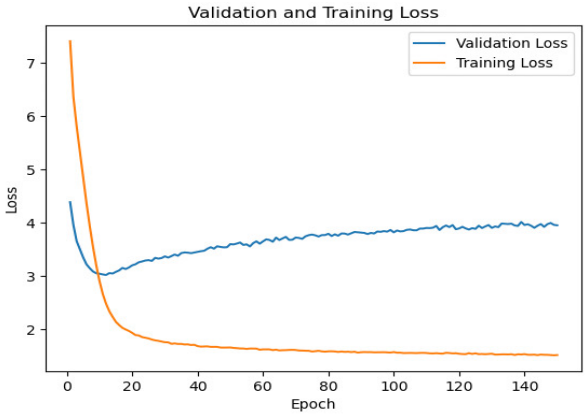
Table 1: Accuracy Table

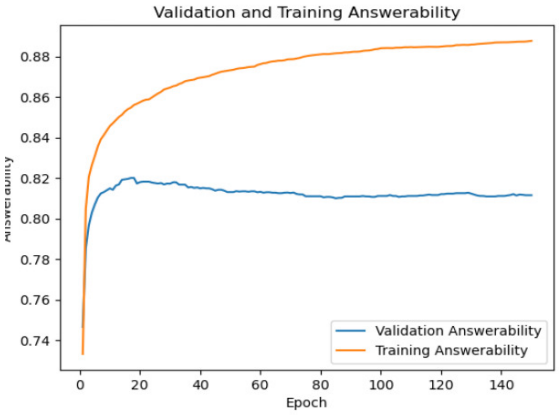
Training	Validation	Testing
81.7%	36.7%	42.1%

Table 2: Answerability Table

Training	Validation	Testing
88.4%	81.1%	84.3%







3 Conclusion

We got less 18% than the stated paper when we implemented the model, however we got better answerability. This model provides a verstile flexiable not taking too long to train. The implemention of answerability was not stated in the paper so we learned it from their demonstration video uploaded on Youtube.