1 Visual Question Answering (VQA)

1.1 Introduction

Visual Question Answering is a task that combines concepts from the world of Computer Vision as well as Natural Language Processing. The idea behind the challenge is to train machines to be able to develop vision like humans so that they can be used to assist people with vision impairments. In this experiment, we have trained three deep neural networks using Tensorflow Keras to be able to predict an object using an image and a question as input. The dataset used for training is the VizWiz - VQA [1] which has over 20000 training images. For the purpose of this experiment, we have used a scaled-down version of this data with 2000 training images and 500 validation images. The VGA task is treated as a classification task for the purpose of this experiment.

1.2 Methods

The software platform used to perform this task is Google Colab with GPU and the hardware used is a Macbook Pro. Since every image corresponds to 10 annotated labels, we have taken the most frequent label out of those as the target output. Feature extraction is performed on the image and the question separately. For the image, we have used the VGG16 model available via TensorFlow Keras. This model has been pretrained on the imagenet data and can be applied to any images to extract features. We reshaped our images to 240 x 240 in order to get the predictions from the vgg16. For the textual questions, we have used a binary encoding to extract the features. This involves creating a vocabulary of words across the entire training dataset. For each input, the output after this encoding will be a vector of the size of words in the vocabulary. If the word is present in the input, its position in the feature vector will be 1, otherwise 0. One-hot encoder provided by sklearn is used to encode the labels using the get_dummies function available opensource.

The first model is a deep neural network comprising 3 layers. The first layer is a fully-connected layer with 10 neurons and the input shape same as the size of the combined feature vector (31893,). The second layer is a fully-connected layer with 12 neurons. Following is the output layer with 1409 neurons to represent the 1409 labels in the dataset. The relu activation function has been used for all the layers but the final. The softmax activation function has been used for the output layer since it promises good performance in the case of multiclass classification tasks. The model is compiled using a binary cross-entropy loss function and optimized using adam optimizer.

The second model contains 4 fully connected layers. The input layer has 100 neurons and the same input shape as the combined feature vector (31893,). The subsequent fully connected layers

have 64 and 10 neurons respectively. The relu activation function is used for all layers but the output. The output layer is a dense layer with 1409 neurons to represent the class labels and softmax activation is used. The model is compiled using a binary cross-entropy loss function and optimized using adam optimizer.

The third model contains an input dense layer containing 100 neurons and an input shape the same as the combined feature vector (31893,). Following this is a dense layer with 12 neurons and the relu activation function. We have then added a dropout layer with the size of 0.2 followed by the final layer is a fully-connected layer with 1409 neurons to represent the class labels and a softmax activation function. The model is compiled using a binary cross-entropy loss function and optimized using RMSprop optimizer.

The prediction for the validation set is calculated by taking the argmax of all the softmax probabilities for each label for every input. This argmax gives us the column index which is then used to decode the one-hot encoding and get the predicted label.

The initial extraction and training process led to an accuracy of 49% for model 1, 15% for model 2, and 45% for model 3. However, these accuracies seemed extremely inconsistent given the similarity of the three models. Upon closer inspection, we discovered that the models were only predicting four output labels: ‘unanswerable’, ‘unsuitable’, ‘yes’, and ‘no’. This happened due to the imbalance in the training dataset whose majority comprised these 4 labels. In order to evaluate the models fairly, we decided to extract training data again leaving out the ‘unanswerable’ and ‘unsuitable’ tagged images. The new split contained of 2177 training samples and 545 validation samples. The batch size was constant for training as the default 32. Each model was retrained on this new dataset and the results are tabulated below.

Model	Number of training images	Number of epochs	Optimizer	Train Accuracy	Val Accuracy	VQA Accuracy
1	2177	10	Adam	12.54%	13.39%	51%
2	2177	10	Adam	14.15%	13.58%	48%
3	2177	20	RMSprop	12.59%	13.58%	52%

1.3 Performance and Analysis

According to the VQA accuracy metric on the validation set, the best-performing model is Model 3. We have used the existing model trained on 2177 samples as the final model. The training and validation loss, as well as training and validation accuracy, are monitored throughout the training. The findings are depicted graphically below.

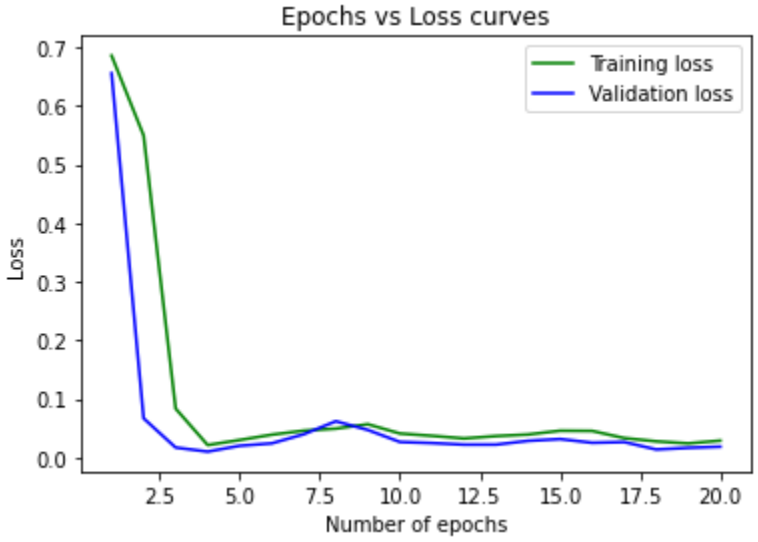

Fig 1: Epochs vs Loss Curve

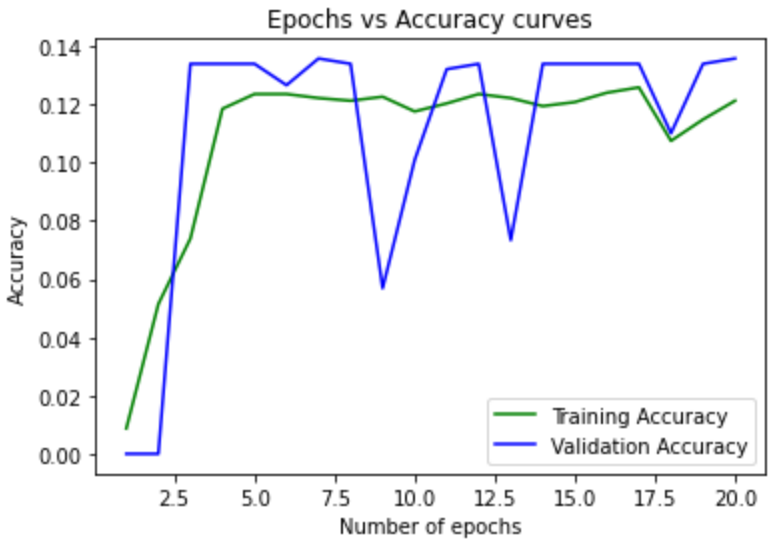

Fig 2: Epochs vs Accuracy Curve

After re-extracting and training leaving out the unanswerable and unsuitable labels, we still observe a sharp bias towards the 'unsuitable image', 'yes' and 'no' labels. This happens due to the imbalance in the dataset which needs to be handled via further preprocessing. Out of the three models trained, the model with the Dropout and RMSprop optimizer worked best with the given dataset. We believe this happens due to the addition of the dropout layer which helps reduce overfitting on the training dataset. The role of the RMSprop optimizer in the performance of the final model remains unclear as the Adam optimizer combines RMSprop with momentum and should ideally perform better in our situation.

The number of epochs was increased during the last training model, however, we believe there is a small but significant role played by the epochs in the final output. This is due to the fact that the training and validation loss reduce rapidly for the first 5 epochs after which they oscillate between 0.0 and 0.1. Whereas, the training and validation accuracy increase sharply till the 5th epoch and then increase slowly with quite a few dips throughout the training process.

The loss curve shows that the training after 5 epochs has little effect on the fitting of the model. The model seems to have been fitted with few epochs and addition of epochs might lead to overfitting in the future. Till the 20th epoch, the model does not seem to overfit since the validation accuracy is higher than the training accuracy and the validation loss is lower than the training loss.

The final model that was used on the test data is the same as model 3. It contains a dense layer as input containing 100 neurons and an input shape the same as the combined feature vector (31893,) and the relu activation function. Following this is a dense layer with 12 neurons and the

relu activation function. We have then added a dropout layer with the size of 0.2 followed by the final layer is a fully-connected layer with 1409 neurons to represent the class labels and a softmax activation function. The model is compiled using a binary cross-entropy loss function and optimized using RMSprop optimizer. The batch size is the default (32) and the number of epochs trained is 20.

1.4 Conclusion

The best model trained is far from capturing the ground truth in the dataset. The presence of unanswerable questions in the dataset makes the training task harder. However, with proper preprocessing, we might be able to improve the performance of our model. Another way of improving performance might be to delve into treating this problem as a generative task as opposed to a classification task. Due to the large number of labels to be predicted, it seems like a generative approach might lead to better and more scalable results.

References

[1] https://vizwiz.org/tasks-and-datasets/vqa/
[2] VizWiz Grand Challenge: Answering Visual Questions from Blind People
Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[3] https://sanjayasubedi.com.np/nlp/nlp-feature-extraction/
[4] https://medium.com/@mygreatlearning/what-is-vgg16-introduction-to-vgg16-f2d63849f615
[5] Coding Tutorial 10 - Samreen Anjum