Linear Layer on CLIP model as powerful vizmiz model paper Implementation

Abdelrhman Elnainay et.all

June 2023

1 abstract

CLIP based Architecture were successfully used in the VQA -visual question answering- models to address the problem of the complexity of these model and the high computational power that they require as mentioned by the Linear Layers on CLIP Features as Powerful VizWiz Model[1]. This paper provides Implementation for such intelligent Architecture using pytorch and CLIP APIs achieving better AP score 84.2% in Task 2: Predict Answerability of a Visual Question than the paper performance 83.78% and close accuracy 54.2% in Task 1: Predict Answer to a Visual Question to that introduced in the paper 60.15% for a network trained for only 2.5 hours and 5 epochs!

2 introduction

Visual Question Answering (VQA) is the task of answering open-ended questions based on an image. VQA has many applications: Medical VQA, Education purposes, for surveillance and numerous other applications. As a result introducing a simple usable Architecture that can be trained using less resources could be highly beneficial in the sense of serving the critical mentioned applications and people. The approach mentioned in the paper was using CLIP pretrained image-text encoder trained on 400 million image-text pairs with a constructive objective to bring both modalities into the same embedding space. The CLIP model was used as a feature extracting layer at the start of the network while totally freezing its parameter, a simple classifications layer at the head have the only parameters to be trained. The task in the paper was carried out on the vizmiz dataset which have several issues in that it has several issues in the data. Questions may not be answerable due to missing information in the images or the quality of the images may be extremely poor. Additionally the questions in the data set are not developed with a rigid set of rules, but are often colloquially. The vizmiz dataset contains image/question pairs with 10 ansewrs for each image the total number of samples was 20523 for the training and about 4319 for validation

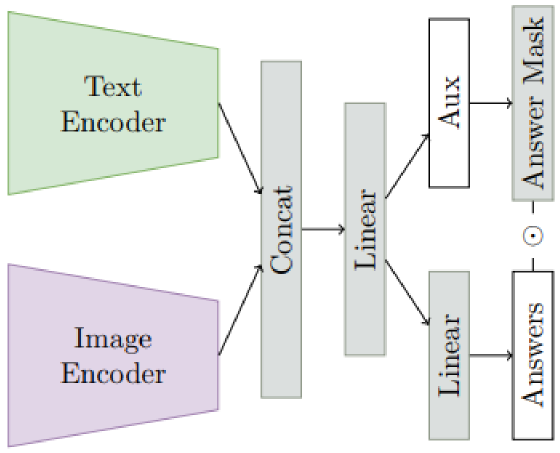


Figure 1: CLIP Architecture for VQA model in fabian.deuser et all paper

3 implementation

the implementation in the paper was carried out in several steps and we wil discuss the implementation details for each step using pytorch.

(i) creating a suitable vocabulary for the classification task, This step was carried by looping on the answer of each question selecting the answer with vast majority only from the answers whose confidence value was yes the classification vocabulary was formed like wise generating a vocabulary space with each vector having 5860 element.

(ii) using CLIP features with linear layers for VQA, The CLIP features were used as feature extracting layer. we used the CLIP API to use those layers

```
import clip
model, preprocess = clip.load(clipType,device)
image_features = model.encode_image(image)
text_features = model.encode_text(text)
```

After that the image and text features are concatenated

```
combined_features = torch.cat((text_features, image_features
    ), dim=-1).to(self.device).to(torch.float32)
```

then the concatenated features are introduced to a fully connected layer with dropout = 0.5 as mentioned in the paper

```
x = self.ln1(combined_features)
x = self.dp1(x)
```

```
x = self.fc1(x)
```

after that two branches are introduced, one to predict the answer and ther other to predict the answer type
answer prediction branch

```
ans = self.ln2(x)
ans = self.dp2(ans)
vqa = self.fc2(ans)
```

(iii) introducing an answer type gate to create a learnable masking. answer type prediction branch with learnable masking layer

```
aux = self.fc_aux(x)
gate = self.fc_gate(aux)
gate = self.act_gate(gate)
```

then the final answer is the attention of the masked aux with the vqa

```
output = vqa * gate
```

further more we added a branch to predict the answerability of the model

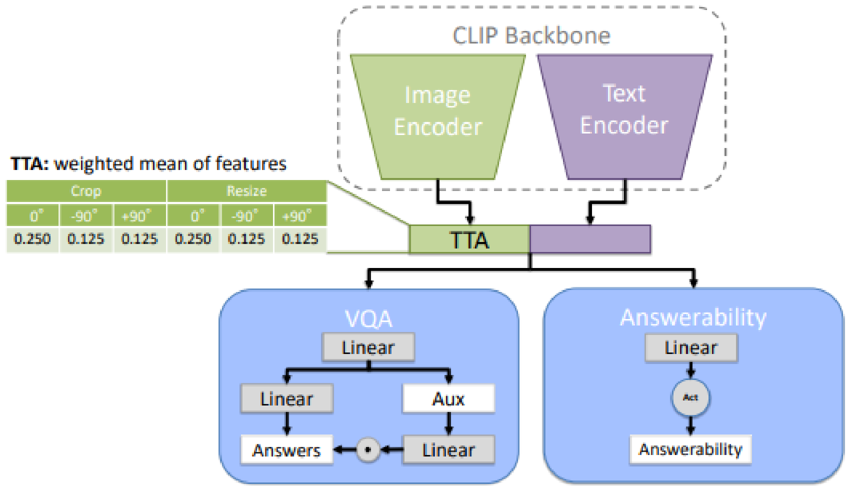


Figure 2: total clip model with answerability branch

```
ansable = self.fc3(x)
ansable = self.act3(ansable)
```

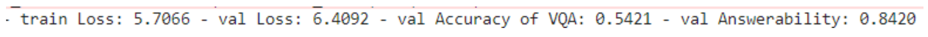
And then the output of the network returns the answer, the answer type and the answerabilty of the question

```
return output, aux, ansable
```

lastly the used loss function is the cross-entropy loss as mentioned in the paper and optimizer Adam with learning rate (lr = 0.01)

```
loss_function = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(vqaModel.parameters(), lr
    =0.001)
```

The training was done for the two types of clip models mentioned in paper "ViT-L/14@336px" and "RN50x64" with 10 epochs for each model. The total time for training each model was about 6 hours each. with best accuracies achieved at the first 3 hours of the training.



```
- train Loss: 5.7066 - val Loss: 6.4092 - val Accuracy of VQA: 0.5421 - val Answerability: 0.8420
```

Figure 3: Best achieved accuracy during training using "ViT-L/14@336px"

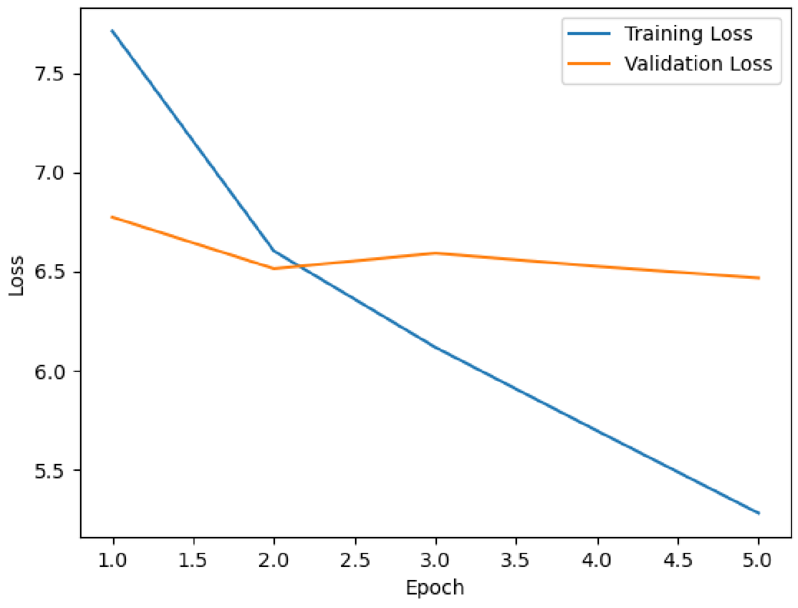
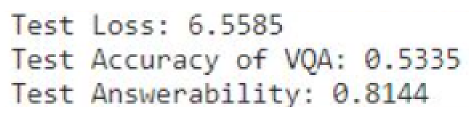
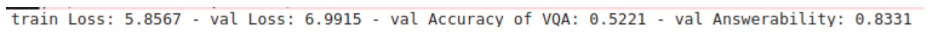


Figure 4: Training and validation loss graph for "ViT-L/14@336px"



```
Test Loss: 6.5585
Test Accuracy of VQA: 0.5335
Test Answerability: 0.8144
```

Figure 5: Test Accuracy using ”ViT-L/14@336px”



```
train Loss: 5.8567 - val Loss: 6.9915 - val Accuracy of VQA: 0.5221 - val Answerability: 0.8331
```

Figure 6: Best achieved accuracy during training using ”RN50x64”

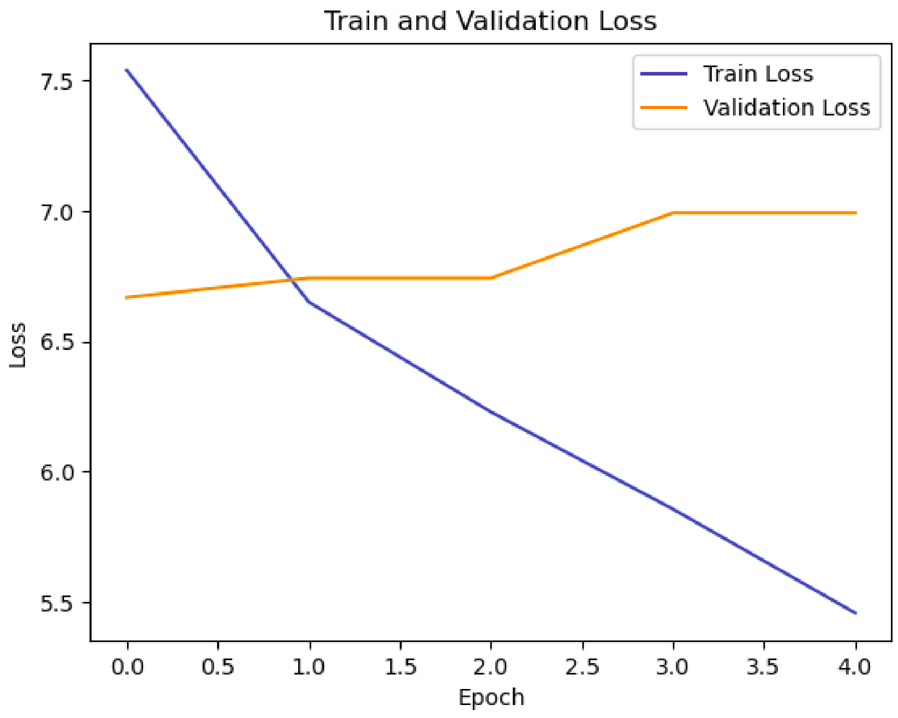
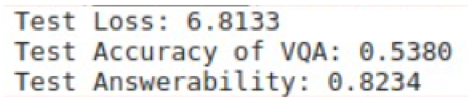


Figure 7: Loss Curve During Training using ”RN50x64”



```
Test Loss: 6.8133
Test Accuracy of VQA: 0.5380
Test Answerability: 0.8234
```

Figure 8: Test Accuracy using ”RN50x64”

provided the link of the notebook for complete code VQA$_clip_model_paper_implementation.com$

4 Conclusion

(1) Pre-trained models can be efficiently used for tasks like VQA (Visual Question Answering), where they reduce the need for additional resources and simplify network architecture.
(2) In our implementation of the paper, we achieved 84.2% accuracy for the answerability task and 54.2% accuracy for the visual question answering task using a network trained for only 2.5 hours and 5 epochs.
(3) The CLIP pre-trained model is very efficient for extracting features from both image and text inputs.

References

[1] Philipp J. Rosch Norbert Oswald Fabian Deuser, Konrad Habel. Less is more: Linear layers on clip features as powerful vizwiz model, Jun 2022.