Hybrid Sequential Vision Transformer

Omer Tafveez - 25020254

1 Overview

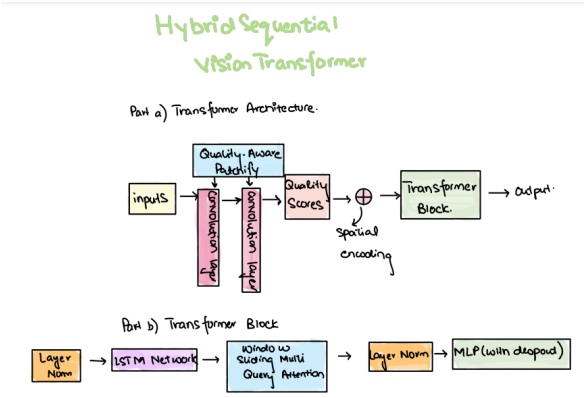

Figure 1: Hybrid Sequential ViT Architecture

This novel approach combines the power of Simple ViT, and produces a Vision Transformer with improved image pre-processing, separate block for local attention, and state-of-the-art Multi-Headed Architecture being currently used in famous Large Language Models such as Mistral, and GPTs.

After being trained on TinyImageNet Dataset, it stands at 85% accuracy, with an f1 score of 0.84, leading to remarkable results in just 5 epochs.

Note: The dataset was truncated to 50k samples to ensure faster computational training due to the lack of GPU and Units on Google Colab.

2 Architecture

2.1 Data Pre-Processing with Quality-Aware Scores

Instead of patchifying the images and flattening them directly, we inserted two convolutional layers in order to extract spatial information and features more conveniently. This ensured more information was passed to the Transformers block.

It is evident in past studies and in the previous ViT Architectures that passing on higher-level features with more information ensures better training.

Furthermore, the difference in resolution/quality of the images is a persistent problem in image classification. Therefore, we appended quality-aware scores in the patches that were passed in the positional embeddings to favor images that had higher quality. The quality estimation is used to modulate the features extracted by the primary convolutional layers of the patchy module. This is done by dot product between the features extracted and the qualities scores obtained from sigmoid function (between 0-1) to obtain patches that were proportional to their scores. This meant that transformers relied more on images for training that had higher quality.

Lastly, the convolved images with quality attention scores are sent to spatial encoding for positional embedding using the same conventional method.

2.2 LSTM Network

Self Attention has been a breakthrough in training longer dependencies and solving information bottlenecks by passing in all hidden states with attention scores. However, there is some evidence that in sequential images or longer text, the self-attention layer loses local attention (context), in broader self-attention, whereas having a supplemental component such as RNN or LSTM network can ensure capturing different patterns of data using a Hybrid Learning (recurrent and self-attention). This is one of the main foundation of our model that proves through results that a hybrid approach to learning has underlying benefits to problems that transformers sometimes cannot deal with on its own.

2.3 Sliding Window Multi Query Attention Block

There is a common debate that self-attention is linear, so we add an MLP or FCNN layer to add non-linearity. Adding non-linearity helps capture the information in the data within the transformer block by adding non-linear activation functions. However, despite adding non-linear activations the operations in self-attention have a time complexity of $O(n^2)$. Latest Popular Large Language Models, such as the family of GPT and Mistral, use "***Sub Quadratic Attention Heads.***". Since 2017, Patch Attention, Flash Attention, and Sliding Window Multi-Query Attention have brought computational expenses down by a huge margin, especially in Memory usage, with time complexity of $O(n)$.

2.4 Multi-Layer Perceptron and Encoder

Using the same conventional Encoder, we implemented a little complex MLP using dropout layers as a part of Network Compression to reduce Memory computing and as a regularizer to avoid overfitting. Dropout has been a revolutionary change in MLP to avoid co-adaptation neurons with each other due to feed-forward and backpropagation operations. A Layer Norm follows this to Normalize and help regularize the model.

3 Results and Training Cycle

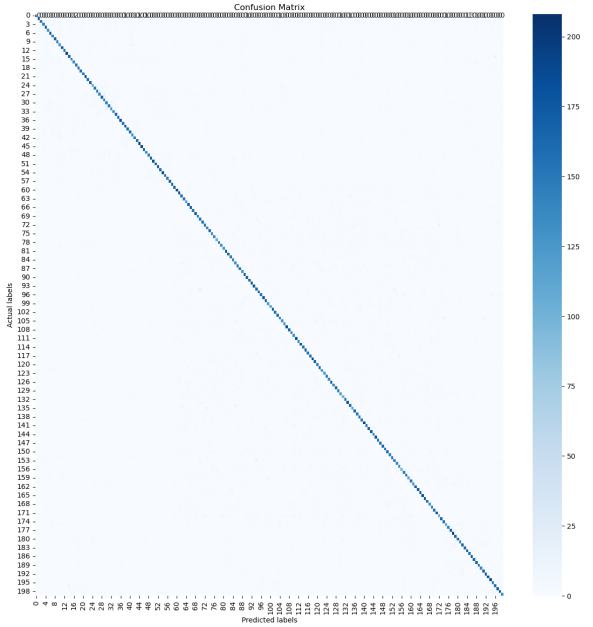

Figure 2: Confusion Matrix at Epoch 1

The above confusion Matrix was obtained with the accuracy of 78% Accuracy.

The training cycle involved 5 epochs with a learning rate of 0.0001. We used AdamW optimizer with a schedular."**OneCycleLr**"

OneCycleLr schedular starts from the lower bound Learning rate, which was defined as 0.0001 up to the upper bound of 0.001 and then starts decreasing it. In cases of image classification of images having RGB channels, OneCycleLr converges to the local minima faster and has a better performance overall.

AdamW is a variant of Adam optimizers that decouples weight decay in optimization which leads to better generalization and provided stable weight-decay operation by applying weight decay to parameters.

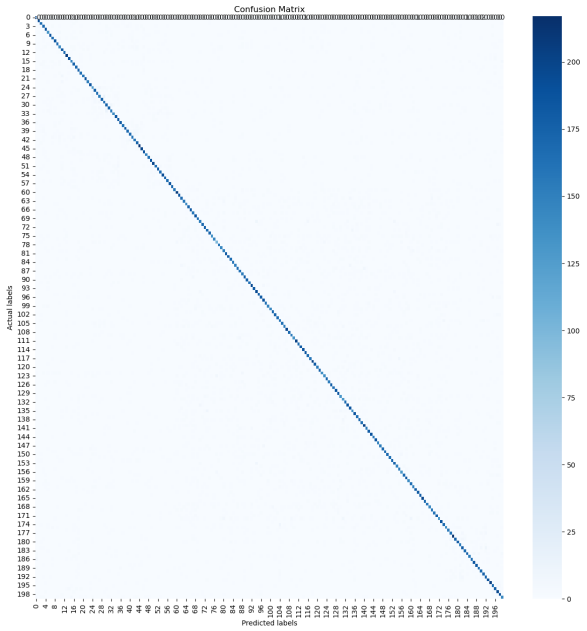

Figure 3: Confusion Matrix at Epoch 5

Epoch	Accuracy	F1-Score	Training Loss
1	0.7848	0.7842	0.9667
2	0.791	0.7901	0.899
3	0.7767	0.7755	0.8504
4	0.8165	0.8155	0.7164
5	0.8502	0.8494	0.6056

Figure 4: Statistics for each Epoch