EAI 2024 Fall

Paper Review Assignment 1

# Review and Analysis of "An Image is Worth 16X16 Words: Transformers for Image Recognition at Scale"

學生：施宇庭 NN6124030

指導教授：蔡家齊 助理教授

# 1 Motivation

Transformers have achieved tremendous success in the natural language processing (NLP) field in recent years, thanks to their model design that gives them high scalability. As the dataset scale grows, the model has not yet reached its performance bottleneck.

In the computer vision (CV) field, some research has been trying to combine CNNs with self-attention, while others has completely replaced the convolution layers with self-attention. However, due to the specially designed attention layers not scaling effectively enough, ResNet-like architectures still remained the state-of-the-art (SOTA) for image recognition prior to this paper.

Therefore, this paper aims to address this issue by splitting the image into small patches and input them into the transformer, just like how NLP takes a sequence of words as input. This approach tries to stay as close as possible to the original transformer design, in order to exploit its efficiency and scalability on modern hardware.

# 2 Proposed Method

## 2.1 Model Architecture

The model architecture of Vision transformer (ViT) is shown in Figure 1.

To get the input of the transformer encoder:

1. Reshape the input image $\mathbf{x} \in \mathbb{R}^{1\times H\times W\times C}$ to a sequence of 2D patches $\mathbf{x}_p \in \mathbb{R}^{N\times P\times P\times C}$, where $N = HW/P^2$ is the number of patches, and $P$ is the patch size.
2. Flatten and project each patch $\mathbf{x}_p^i$ to $\mathbf{x}_p^i\mathbf{E} \in \mathbb{R}^D$, where $\mathbf{E} \in \mathbb{R}^{(P^2\cdot C)\times D}$
3. Prepend a learnable class token as $\mathbf{x}_p^0$ to the sequence, and plus the positional embedding $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1)\times D}$.

This forms the patch embeddings (Eq. 1).

$$\mathbf{z_0} = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; ...; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{\text{pos}} \tag{1}$$

The transformer encoder consists of alternating layers of multihead self-attention (MSA) and multi-layer perceptron (MLP) blocks as shown in [3]. The difference is that the encoder of this

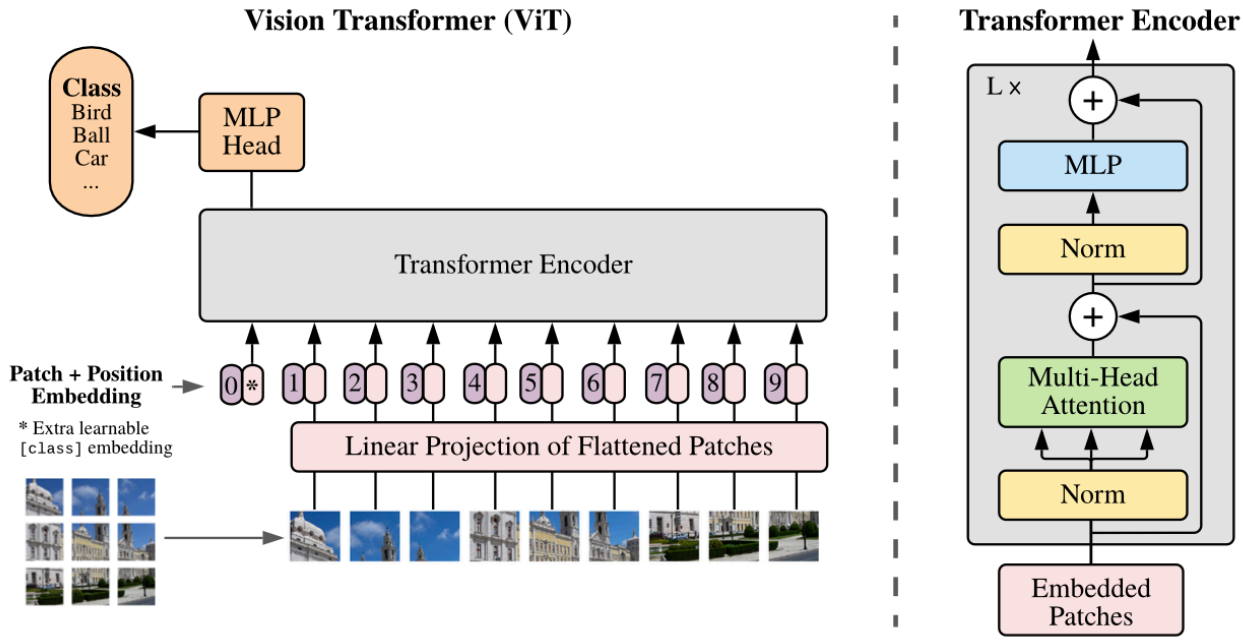


Figure 1: Model architecture of Vision Transformer [1]

paper applies layer normalization (LN) before every block, and residual connections after every block like the Pre-LN archtecture in [4].

$$\mathbf{z'}_l = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{l-1})) + \mathbf{z}_{l-1}, \qquad l = 1...L \tag{2}$$

$$\mathbf{z}_l = \mathrm{MLP}(\mathrm{LN}(\mathbf{z'}_l)) + \mathbf{z'}_l, \qquad l = 1...L \tag{3}$$

$$\mathbf{y} = \mathrm{LN}(\mathbf{z}_L^0) \tag{4}$$

Note that the MLP contains two layers with GELU activation function.

### 2.2 Pre-training and Fine-tuning

This article pre-trains ViT on large datasets, and then fine-tunes to a relatively smaller downstream tasks.

## 3 Experiment and Evaluation

To demonstrate the effectiveness of the proposed method, this work evaluate the ViT with ResNet-like CNN, and the hybrid.

| | Ours-JFT (ViT-H/14) | Ours-JFT (ViT-L/16) | Ours-I21k (ViT-L/16) | BiT-L (ResNet152x4) | Noisy Student (EfficientNet-L2) |
|---|---|---|---|---|---|
| ImageNet | **88.55** $\pm$ 0.04 | 87.76 $\pm$ 0.03 | 85.30 $\pm$ 0.02 | 87.54 $\pm$ 0.02 | 88.4/88.5* |
| ImageNet ReaL | **90.72** $\pm$ 0.05 | 90.54 $\pm$ 0.03 | 88.62 $\pm$ 0.05 | 90.54 | 90.55 |
| CIFAR-10 | **99.50** $\pm$ 0.06 | 99.42 $\pm$ 0.03 | 99.15 $\pm$ 0.03 | 99.37 $\pm$ 0.06 | – |
| CIFAR-100 | **94.55** $\pm$ 0.04 | 93.90 $\pm$ 0.05 | 93.25 $\pm$ 0.05 | 93.51 $\pm$ 0.08 | – |
| Oxford-IIIT Pets | **97.56** $\pm$ 0.03 | 97.32 $\pm$ 0.11 | 94.67 $\pm$ 0.15 | 96.62 $\pm$ 0.23 | – |
| Oxford Flowers-102 | 99.68 $\pm$ 0.02 | **99.74** $\pm$ 0.00 | 99.61 $\pm$ 0.02 | 99.63 $\pm$ 0.03 | – |
| VTAB (19 tasks) | **77.63** $\pm$ 0.23 | 76.28 $\pm$ 0.46 | 72.72 $\pm$ 0.21 | 76.29 $\pm$ 1.70 | – |
| TPUv3-core-days | 2.5k | 0.68k | 0.23k | 9.9k | 12.3k |

Figure 2: Comparison with SOTA CNNs on popular image classification benchmarks.

This paper conducts several experiments to investigate the effectiveness and operation of ViT by:

1. Comparing the performance and computational cost of ViT and SOTA CNNs on popular image classification datasets. (Figure 2 and 3)
2. Examining the impact of different pre-training dataset sizes. (Figure 4 and 5)
3. Assessing the scalability of ViT across different sizes. (Figure 6)
4. Analyzing the internal operation of ViT.

## 3.1 Comparison to State of The Art

To demonstrate the effectiveness of the proposed method, this study compares the accuracy and computational cost (TPUv3-cores-days) of ViT with that of SOTA CNNs on popular image classification datasets. Figure 2 shows the results, and we can find that:

- A smaller ViT-L/16 pre-trained on JFT-300M outperforms BiT pre-trained on JFT-300M across all tasks.
- A larger ViT-H/14 exhibits even better performance than ViT-L/16.
- ViTs require fewer computational resources for pre-training compared to other SOTA CNNs.

However, pre-training efficiency is influenced not only by the model architecture but also by other factors such as training schedule, optimizer, and weight decay.

Figure 3 decomposes the VTAB tasks into their respective groups, and compares to previous SOTA methods on this benchmark. ViT-H/14 outperforms BiT-R152x4, and other methods,

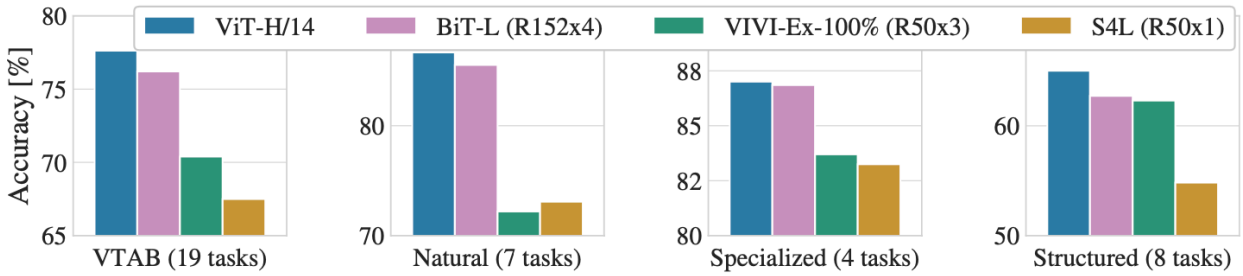


Figure 3: Breakdown of VTAB performance in *Natural*, *Specialized*, and *Structured* task groups.

on the *Natural* and *Structured* tasks. On the Specialized the performance of the top two models is similar.

## 3.2 Pre-training Data Requirements

To investigate how crucial is the dataset size of pre-training, this paper further compared the accuracy of ViT and BiT with varying pre-training dataset sizes: ImageNet, ImageNet-21k, and JFT300M. Figure 4 shows the result after fine-tuning to ImageNet, and the results are:

- When pre-trained on small dataset (ImageNet), ViT-Base models are better than ViT-Large models, despite the regularization including weight decay, dropout, and smoothing. And the BiT model outperforms the ViT models.
- When pre-trained on medium-sized dataset (ImageNet-21k), their performances are similar.
- When pre-trained on large dataset (JFT-300M), larger ViT start revealing the benefit.

An additional experiment was performed on random subsets of 9M, 30M, and 90M as well as the full JFT300M dataset. As shown in figure 5, similar result holds: ResNets perform better with smaller pre-training datasets, while ViT performs better with larger pre-training datasets.

## 3.3 Scaling Study

Observing the transformer performance on JFT-300M with different models allows for an assessment of scalability (performance/compute trade-off). The result is shown in figure 6. We can observe that:

- ViT dominate ResNets on the performance/compute trade-off.

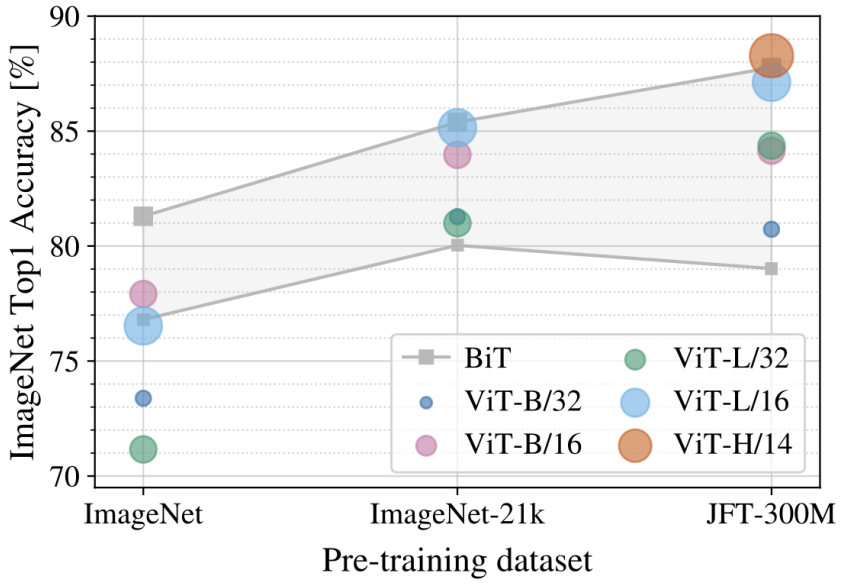


Figure 4: Transfer to ImageNet.

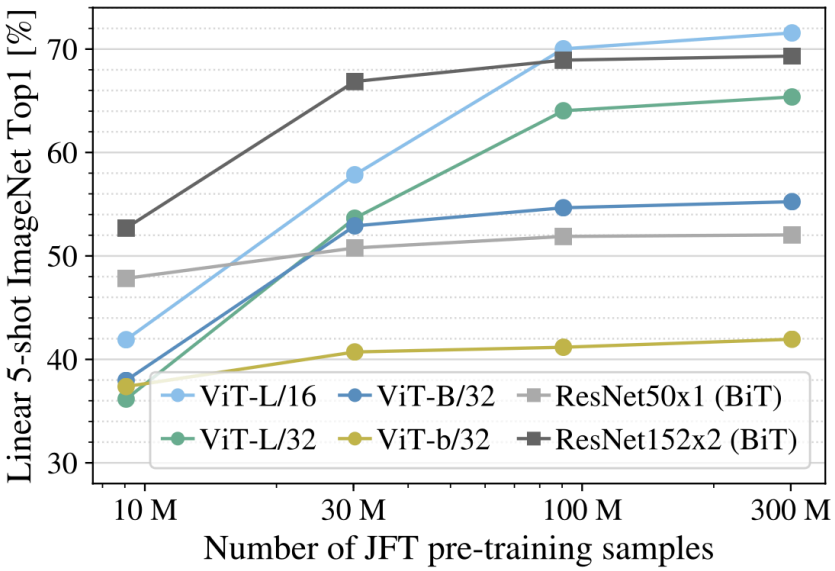


Figure 5: Linear few-shot evaluation on Ima- geNet versus pre-training size.

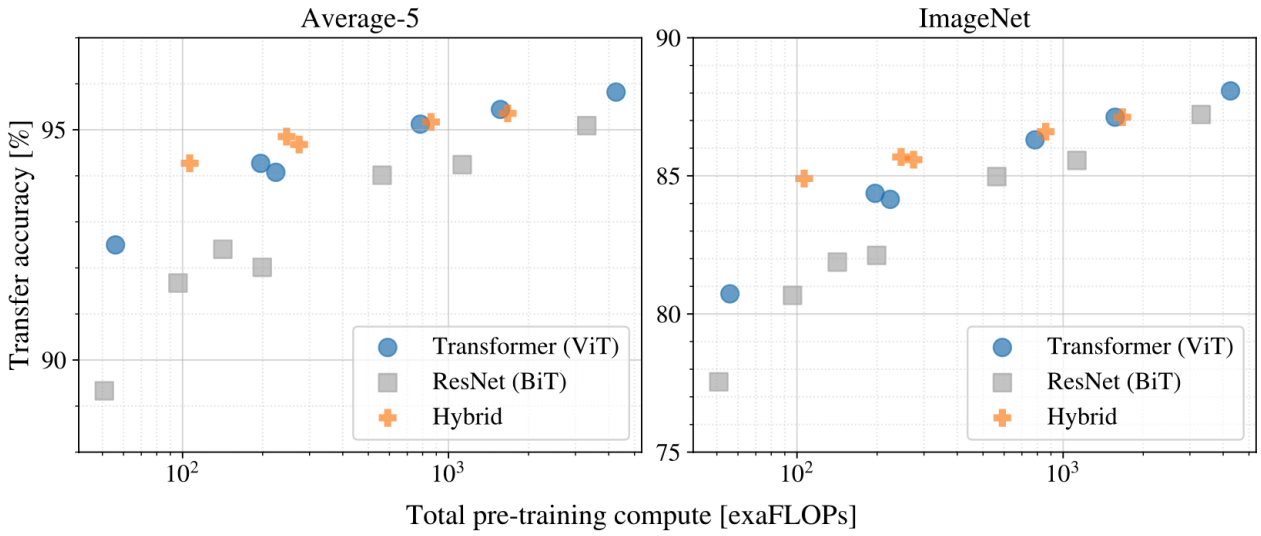


Figure 6: Performance versus pre-training compute for different architectures: Vision Transformers, ResNets, and hybrids.

- hybrids slightly outperform ViT at small computational budgets, but the difference vanishes for larger models.
- Vision Transformers appear not to saturate within the range tried, motivating future scaling efforts.

# 4 Analysis

## 4.1 Pre-LN v.s. Post-LN Transformer

When reviewing the relevant literature, I noticed that the encoder architecture proposed in this paper differs slightly from the original transformer architecture. The original model architecture of transformer proposed by [3] is shown in figure 7. Comparing with the figure 1, we can observe that the relative position of the layer normalization differs. In the original paper, the LN layer comes after the MSA and MLP, while in this paper, the LN layer comes before the MSA and MLP. In this paper, the rationale behind such a design choice is not explicitly discussed. However, a study by [4] suggests that in the original paper, the Post-LN approach results in larger gradients near the output layer, leading to instability during training with larger learning rates. Consequently, an additional warm-up stage is required, introducing more variability in experiments (unlike the training procedure of CNNs). In contrast, employing Pre-LN results in smaller gradients, making it more suitable for deep neural networks and eliminating the need for a warm-up stage.

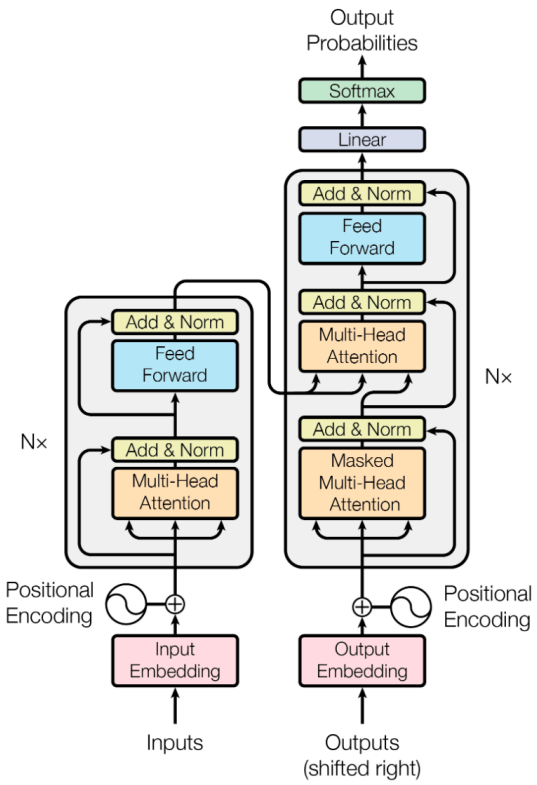


Figure 7: The original model architecture of transformer proposed by [3]

## 4.2 Workload Analysis

When deploying the ViT model in a real-world scenario, or designing an optimized hardware/software co-design system, it is essential to understand the workload characteristics of the model. The workload characteristics of the ViT model can be analyzed from the following aspects:

- **Computation Requirements:** The computational requirements for each layer, including the calculation process for each operator, with metrics expressed in **FLOPs**.
- **Memory Requirements:** The memory requirements for each layer, assuming that inputs and outputs of all operations are fetched from and stored to DRAM, with metrics expressed in **Bytes**.

There are three variants of the ViT model: ViT-Base, ViT-Large, and ViT-Huge. Their detailed configurations are shown in Table 1.

The following we will take ViT-Base on ImageNet-1k classification task as an example. The hyperparameters are shown in Table 2. Assume that the input image size is $224 \times 224$ pixels, and the patch size is $16 \times 16$ pixels. The number of patches is 196, and the embedding size is 768. The number of layers is 12, and the number of heads is 12. The number of classes is

| Model | Layers | Hidden size | MLP size | Heads | Params |
|---|---|---|---|---|---|
| ViT-Base | 12 | 768 | 3072 | 12 | 86M |
| ViT-Large | 24 | 1024 | 4096 | 16 | 307M |
| ViT-Huge | 32 | 1280 | 5120 | 16 | 632M |

Table 1: ViT model variants [1]

| Hyperparameter | Value |
|---|---|
| $b$: batch size | 1 |
| $N$: sequence length (number of patches) | $\frac{224}{16} \times \frac{224}{16} = 196$ |
| $d$: embedding size, hidden size | $3 \times 16 \times 16 = 768$ |
| $V$: number of output classes | 1000 |
| $a$: number of heads in MSA block | 12 |
| $l$: number of encoder layers | 12 |

Table 2: Hyperparameters for ViT-B.

1000. The computation and memory requirements for each layer can be calculated based on the model architecture and hyperparameters.

### 4.2.1 Multi-head Self-attention (MSA) Block

The MSA block in ViT consists of three main components: Query, Key, and Value projections, scaled dot-product attention, and output projection. The input is normalized by LayerNorm, and the output follows a residual connection. The architecture diagram is depicted in Figure 8. The computational and memory access requirements for each layer in the MSA block are shown in Table 3.

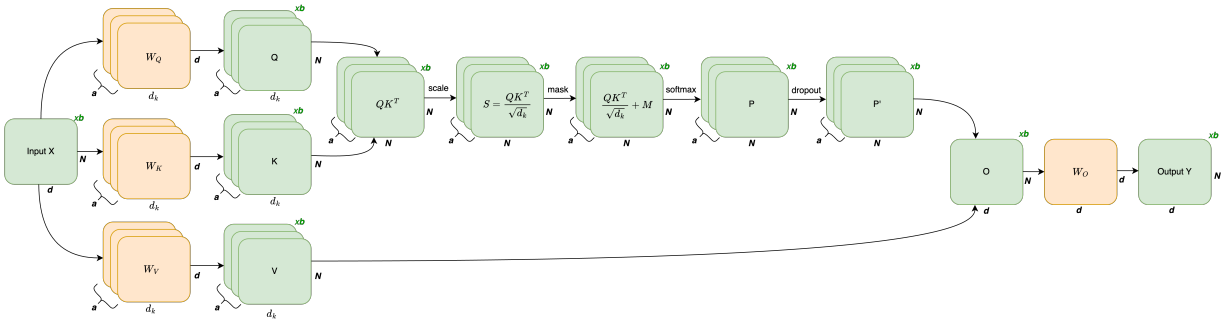


Figure 8: MSA block in ViT.

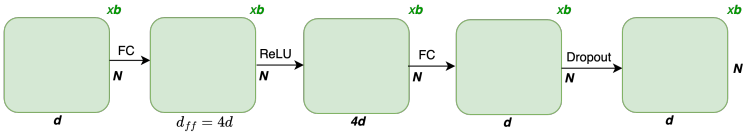


Figure 9: MLP block in ViT.

| Layer Name | FLOPs | Memory Access (Bytes) | Operational Intensity |
|---|---|---|---|
| LN | $6Nbd$ | $8Nbd$ | $\frac{3}{4}$ |
| Q proj | $2Nbd^2$ | $8Nbd + 4d^2$ | $\frac{Nbd}{4Nb+2d}$ |
| K proj | $2Nbd^2$ | $8Nbd + 4d^2$ | $\frac{Nbd}{4Nb+2d}$ |
| V proj | $2Nbd^2$ | $8Nbd + 4d^2$ | $\frac{Nbd}{4Nb+2d}$ |
| QK matmul | $2N^2bd$ | $48Nbd + 4N^2ba$ | $\frac{Nbd}{4bd+2Nba}$ |
| scaling | $N^2b$ | $8N^2ba$ | $\frac{1}{8a}$ |
| softmax | $3N^2b$ | $8N^2ba$ | $\frac{3}{8a}$ |
| SV matmul | $2N^2bd$ | $4N^2ba + 8Nbd$ | $\frac{Nbd}{2.5Nba+4bd}$ |
| O proj | $2Nbd^2$ | $4d^2 + 8Nbd$ | $\frac{Nd}{2d+4.5N}$ |
| Add | $Nbd$ | $12Nbd$ | $\frac{1}{12}$ |

Table 3: Computational and memory access requirements for each layer in MSA block.

### 4.2.2 Multi-layer Perceptron (MLP) Block

The MLP block in ViT consists of two fully connected layers with GELU activation function. The input is normalized by LayerNorm, and the output follows a residual connection. The architecture diagram is depicted in Figure 9. The computational and memory access requirements for each layer in the MLP block are shown in Table 4.

### 4.2.3 Classification Head

The classification head in ViT consists of a fully connected layer followed by a softmax function in training stage and argmax in inference stage. The computational and memory access requirements for the classification head are shown in Table 5.

| Layer Name | FLOPs | Memory Access (Bytes) | Operational Intensity |
|---|---|---|---|
| LN | $6Nbd$ | $8Nbd$ | $\frac{3}{4}$ |
| FC1 | $4Nbd^2$ | $20Nbd + 16d^2$ | $\frac{Nbd}{5Nb+4d}$ |
| GELU | $32Nbd$ | $32Nbd$ | $\frac{1}{8}$ |
| FC2 | $4Nbd^2$ | $20Nbd + 16d^2$ | $\frac{4Nbd}{21Nb+16d}$ |
| Add | $Nbd$ | $12Nbd$ | $\frac{1}{12}$ |

Table 4: Computation and memory access requirements for each layer in MLP block.

| Layer Name | FLOPs | Memory Access (Bytes) | Operational Intensity |
|---|---|---|---|
| Linear | $2bdV$ | $4bd + 4dV + 4bV$ | $\frac{bdV}{2bd+2dV+2bV}$ |

Table 5: Computation and Memory Access for Linear Layer

#### 4.2.4 Conclusion

The computational and memory access requirements, as well as the operational intensity for each layer in the ViT model, can be calculated based on the model architecture and hyperparameters.

Using the roofline model allows for further analysis of ViT's performance bottlenecks to determine if each layer is **compute-bound** or **memory-bound**, which can guide performance optimization and deployment efforts.

## 5 Future Work

The authors proposed some future work for this research including:

1. Applying to other computer vision tasks (e.g. detection, segmentation)
2. Large-scale self-supervised pre-training
3. Further scaling on ViT

In addition to the directions proposed by the authors, I believe there are the following directions for future development:

Transformer models have a vast number of parameters and computations, typically trained and inferred on servers. However, even on servers, there is a demand for model compression to save

computation costs and increase throughput [5]. ViT models used in the field of computer vision are no exception. However, traditional quantization and pruning techniques for CNNs may not be directly applicable to transformer model computations. Therefore, developing quantization algorithms suitable for ViT would be a promising research direction. For example, there is already research on post-training quantization (PTQ) for ViT [2]. However, quantization-aware training (QAT) for ViT remains an unresolved issue due to training oscillations causing instability during the training process.

This paper proposes a redesign of the algorithm to enable transformers to efficiently perform computer vision tasks using existing hardware (TPUv3). It also points out that previous papers have not performed well due to the under-utilization of hardware capabilities caused by specific attention patterns. Furthermore, experimental results indicate that existing large datasets have not reached the limit of Vision Transformer and still have the potential for further scaling up. Therefore, developing a software/hardware co-design system for ViT is also a research direction worth exploring.

# References


[1] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. June 3, 2021. DOI: 10.48550/arXiv.2010.11929. arXiv: 2010.11929[cs]. URL: http://arxiv.org/abs/2010.11929 (visited on 03/15/2024).

[2] Zhenhua Liu et al. *Post-Training Quantization for Vision Transformer*. June 27, 2021. DOI: 10.48550/arXiv.2106.14156. arXiv: 2106.14156[cs]. URL: http://arxiv.org/abs/2106.14156 (visited on 03/18/2024).

[3] Ashish Vaswani et al. *Attention Is All You Need*. Aug. 1, 2023. DOI: 10.48550/arXiv.1706.03762. arXiv: 1706.03762[cs]. URL: http://arxiv.org/abs/1706.03762 (visited on 03/16/2024).

[4] Ruibin Xiong et al. *On Layer Normalization in the Transformer Architecture*. June 29, 2020. DOI: 10.48550/arXiv.2002.04745. arXiv: 2002.04745[cs,stat]. URL: http://arxiv.org/abs/2002.04745 (visited on 03/18/2024).

[5] Yilong Zhao et al. *Atom: Low-bit Quantization for Efficient and Accurate LLM Serving*. Nov. 7, 2023. DOI: 10.48550/arXiv.2310.19102. arXiv: 2310.19102[cs]. URL: http://arxiv.org/abs/2310.19102 (visited on 01/29/2024).