Compression and Regularization of Vision Transformers

by

Rajeev Goel

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2023 by the
Graduate Supervisory Committee:

Yingzhen Yang, Chair
Yezhou Yang
Jia Zou

ARIZONA STATE UNIVERSITY

May 2023

ABSTRACT

Vision Transformers (ViT) achieve state-of-the-art performance on image classification tasks. However, their massive size makes them unsuitable for edge devices. Unlike CNNs, limited research has been conducted on the compression of ViTs. This thesis work proposes the "adjoined training technique" to compress any transformer-based architecture. The architecture, Adjoined Vision Transformer (AN-ViT), achieves state-of-the-art performance on the ImageNet classification task. With the base network as Swin Transformer, AN-ViT with 4.1× fewer parameters and 5.5× fewer floating point operations (FLOPs) achieves similar accuracy (within 0.15%). This work further proposes Differentiable Adjoined ViT (DAN-ViT), which uses neural architecture search to find hyper-parameters of our model. DAN-ViT outperforms the current state-of-the-art methods including Swin-Transformers by about $\sim 0.07\%$ and achieves 85.27% top-1 accuracy on the ImageNet dataset while using 2.2× fewer parameters and with 2.2× fewer FLOPs.

DEDICATION

Dedicated to, my loving parents, family and friends who have always believed in me and supported me in all my endeavors.

This work is dedicated to you with love and gratitude.

ACKNOWLEDGMENTS

Working on this thesis was one of the most important things I did during my time as a graduate student. It gave me the chance to learn from professionals, try new things, and enhance my research skills. I take this opportunity to convey my gratitude to all those who helped me directly or indirectly in making this Master of Science thesis possible.

First and foremost, I would like to express my gratitude to my advisor, Dr. Yingzhen Yang, for his unwavering guidance and mentoring. I am grateful for his patience, encouragement, and invaluable insights that have helped shape this work. To my colleagues and friends at SCAI, I extend my heartfelt thanks. The sense of community, support, and intellectual stimulation I've found in this group have helped me a lot as I face the challenges of graduate school. In particular, I would like to thank Mr. Utkarsh Nath for his encouragement and invaluable advice throughout this research work. I appreciate Research Computing at Arizona State University for giving me the tools I needed to carry out my research.

Last but certainly not least, I would like to express my profound gratitude to my family. Their unwavering love, support, and confidence in me have carried me through even the most challenging times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

In computer vision, convolutional neural networks (CNNs) have been the dominant architecture for a long time. Since AlexNet's (Krizhevsky *et al.*, 2017) ground-breaking performance on the ImageNet challenge, CNN architectures have become more potent through increased scale, more connections, and more sophisticated forms of convolution. These developments have resulted in enhanced performance across a vast array of visual tasks.

In contrast, the Transformer architecture has become popular in natural language processing (NLP). Designed for sequence modeling and transduction tasks, transformers use attention to model long-range dependencies in data. Their success in natural language processing (NLP) has led to efforts to use them in computer vision.

Vision transformers (ViTs) achieve state-of-the-art performance for a variety of computer vision tasks such as classification (Dosovitskiy *et al.*, 2021; Liu *et al.*, 2021; Touvron *et al.*, 2021; Yuan *et al.*, 2021), object detection (Li *et al.*, 2022; Liu *et al.*, 2021), and segmentation (Strudel *et al.*, 2021; Xie *et al.*, 2021). In (Dosovitskiy *et al.*, 2021), the researchers introduced transformers for classification tasks, where input image is fed to the transformer as a sequence of small patches for high classification performance on ImageNet. However, it required pre-training on large-scale datasets such as JFT-300M and suffered from huge model sizes. To alleviate these issues, T2T-ViT (Yuan *et al.*, 2021) designed a token-to-token module that transforms the input patches such that the local structure is maintained. DeiT (Touvron *et al.*, 2021) used knowledge distillation to train ViTs from scratch without extensive pre-training. Apart from them, several modified ViTs outperform CNN models and achieve SOTA

performance on the ImageNet dataset (Wang *et al.*, 2021; Chu *et al.*, 2021; Wu *et al.*, 2021; Han *et al.*, 2021).

1.1 Motivation

Despite achieving tremendous success, vision transformers demand much more resource than CNNs, making them difficult to be deployed on edge devices such as mobile phones and embedded devices. The majority of works on building efficient models is based on CNNs (Wu *et al.*, 2019; Liu *et al.*, 2019; Lin *et al.*, 2020b,a). However, compressing ViTs while maintaining state-of-the-art performance is of paramount importance. Standard techniques for compression of vision transformers include pruning, designing efficient transformers, and using neural architecture search.

In this thesis work, we propose Adjoined Vision Transformers (AN-ViT), a training paradigm that can compress arbitrary transformer-based neural architecture. ViTs compressed by AN-ViT achieve SOTA performance on the ImageNet dataset with significantly smaller parameter number and FLOPs. Using Swin-transformers as the base network, AN-ViT achieves up to 85.05% accuracy on the ImageNet dataset with $4.1\times$ fewer parameters and $5.5\times$ fewer FLOPs. AN-ViT outperforms similar sized model by approximately $2.5\% - 3.5\%$ (see Figure 4.1) top-1 ImageNet accuracy. We further propose Differentiable Adjoined ViT, or DAN-ViT, that uses neural architecture search to find hyper-parameters of AN-ViT. DAN-ViT further increases the accuracy of ViTs compressed by AN-ViT, and ViT model compressed by DAN-ViT exceeds the accuracy of base ViT by 0.07%. In Figure 4.1, we compare top-1 ImageNet accuracy of compressed models to the current SOTA and efficient ViTs in terms of model size and FLOPs. It can be observed that models compressed by adjoined training always enjoy higher accuracy and less parameter number and FLOPs compared to base models.

The AN-ViT training paradigm trains the base and the compressed model together. It works as follows. The input image, X, is passed through both the base (or large) model and the compressed (or small) model. AN-ViT produces two probability vectors p and q corresponding to the large and small models, respectively. While training AN-ViT, we force the output of the smaller model (q) to approximate the output of the larger model (p) so as to preserve the prediction accuracy of the compressed model. This setting is similar to Knowledge Distillation (Hinton *et al.*, 2015), where the student model uses the output of a pre-trained teacher model as soft labels to train itself. However, there are two crucial distinctions. First, the weight of the compressed (small) model is a subset of the base (large) model, that is, in AN-ViT, all parameters of the smaller model are shared between the small and the large model. Second, in AN-ViT, we train both the small and large models together, whereas in the teacher-student setting, the parameters of the teacher are fixed, and the student learns from the output of a fixed teacher model.

AN-ViT uses compression factor α to determine the number of heads in each transformer block for the smaller model. In AN-ViT, α is not a learnable parameter and is chosen manually prior to training and a common α is shared by all transformer blocks. However, different blocks capture different features, therefore, they should be compressed using different compression ratios. To this end, we propose DAN-ViT which employs Neural Architecture Search (NAS) to search for the optimal compression factor for each transformer block. Due to the adaptive compression ratio for different transformer blocks, DAN-ViT further improves the model's performance. As illustrated in Figure 4.1, the compressed model by DAN-ViT achieves 85.27 % top-1 accuracy on ImageNet, exceeding the base model by 0.07 % while being 2.27× smaller and 2.2× faster.

1.2 Contributions

Below are the main contributions of this work.

1. We propose Adjoined Vision Transformers (AN-ViT), a training paradigm that can compress arbitrary tranformer based neural architecture. We further propose Differentiable Adjoined ViT, or DAN-ViT, that uses neural architecture search to find hyper-parameters of AN-ViT. DAN-ViT further increases the accuracy of ViTs compressed by AN-ViT.

2. AN-ViT and DAN-ViT achieve state-of-art performance on the ImageNet dataset. We compare our models compressed by with current SOTA efficient ViTs (Huang *et al.*, 2022; Graham *et al.*, 2021; Liao *et al.*, 2021; Chen *et al.*, 2021a; Zhang *et al.*, 2022; Yang *et al.*, 2021; Yu *et al.*, 2022; Zhu *et al.*, 2021; Chen *et al.*, 2021b). Compared to these similar sized model, AN-ViT achieve $\sim (2.5\% - 3.5\%)$ higher ImageNet accuracy.

Chapter 2

LITERATURE REVIEW

This chapter discusses the history of model compression and the current state of the discipline. I will briefly describe knowledge distillation, pruning, neural architecture search, and efficient architectures. In addition, I will discuss how these methods differ from our own Adjoined Network training method.

2.1 Knowledge Distillation

Knowledge Distillation (KD) refers to the transfer of knowledge from a large model to a small one. (Hinton *et al.*, 2015) proposes a teacher-student model in which the student model is trained using soft targets from the teacher. KD forces the student to generalize, similar to the teacher model. Since (Hinton *et al.*, 2015), various knowledge transfer methods have been proposed. (Romero *et al.*, 2015) used intermediate layer's information from the teacher model to train a thinner and deeper student model. (Peng *et al.*, 2019) proposes to use instance level correlation congruence instead of just using instance congruence between the teacher and student. (Ahn *et al.*, 2019) tried to maximize the mutual information between teacher and student models using variational information maximization. The goal of (Park *et al.*, 2019) is to transfer structural knowledge from teacher to student. (Kim *et al.*, 2018) argues that directly transferring a teacher's knowledge to a student is difficult due to inherent differences in structure, layers, channels, etc., therefore, they paraphrase the output of the teacher in an unsupervised manner, making it easier for the student to understand. (Zhang *et al.*, 2017) used an ensemble of students to learn collaboratively and teach each other throughout the training process, rather than using a teacher to train a student. (Yim

et al., 2017) involves distilling knowledge from a pre-trained Deep Neural Network (DNN) and transferring it to another DNN by computing the inner product between features from two layers. In (Zagoruyko and Komodakis, 2017) student network is forced to mimic the attention maps of a teacher model. (Li *et al.*, 2019) combines an asymmetric dual-model learning framework with an intermediate layer selection scheme to identify the corresponding intermediate layers of source and target models. (Tian *et al.*, 2020b) proposes a method for transferring representational knowledge from one neural network to another by distilling a large network into a smaller one, transferring knowledge from one sensory modality to another, or combining multiple models into a single estimator. Most of these methods use a trained teacher model to train a student model. In contrast, in this work, we train both the teacher and the student together.

2.2 Compression of VIT

In this section, we discuss various techniques for compression of VITs such as pruning, neural architecture search, and efficient architectures, and how they differ from our methodology.

2.2.1 Pruning

Pruning techniques aim to reduce the size of a network while maintaining accuracy by removing parameters or weights based on some heuristic. These techniques can be classified into two categories: unstructured pruning and structured pruning. Unlike structured pruning techniques, unstructured pruning approaches are agnostic to the underlying network topology. These methods induce sparsity according to some predefined criteria and frequently achieve a state-of-the-art reduction in the number of parameters. Due to the unstructured nature of these methods, they are frequently

incapable of providing inference speedups on commodity hardware. Unstructured sparsity has been extensively studied in (Evci *et al.*, 2019; Kusupati *et al.*, 2020; Gale *et al.*, 2019; Zhu and Gupta, 2017; Han *et al.*, 2015). Structured pruning address the slow inference times by taking network architecture into consideration. Some recent works, such as (Zhu *et al.*, 2021), promote dimension-wise sparsity in order to reduce the number of heads or MLP hidden dimensions. (Chen *et al.*, 2021b) prunes the network using Taylor importance score. (Yang *et al.*, 2021) improves the compressed models' accuracy by designing a latency-aware structured pruning method that considers all parameters of the VIT. (Yu *et al.*, 2022) moves a step further and unifies pruning, knowledge distillation, and blocking/layer skipping. The AN compression technique proposed in this paper can also be thought of as a structured pruning method where the pruned architecture at each block is fixed at the beginning of the training.

2.2.2 Neural Architecture Search

Neural Architecture Search(NAS) is a technique that automatically designs neural architecture without human intervention. Earlier studies in NAS were based on RL (Zoph and Le, 2017; Tan *et al.*, 2019) and EA (Real *et al.*, 2017); however, they required lots of computation resources. Most recent studies (Liu *et al.*, 2019; Cai *et al.*, 2019; Wu *et al.*, 2019) encoded architectures as a weight-sharing supernetwork. (Chen *et al.*, 2021a; Liao *et al.*, 2021) propose a NAS technique that searches for various changeable dimensions of the transformer, such as embedding dimension, number of heads, query/key/value dimension, MLP ratio, and network depth. Apart from searching (Liao *et al.*, 2021) also propose a residual spatial reduction module to decrease the sequence length and increase embedding dimension for deeper transformer blocks. Most of these techniques focus on designing compact architecture from

scratch. In this work, we use Adjoined training to compress existing architectures. We use architecture search to help us select the compression ratio at each block, i.e., the fraction of heads that should be shared between AN-Large and AN-Small at each block.

2.2.3 Efficient Architectures

Another research direction is to build efficient vision transformers. (Graham *et al.*, 2021) replace the uniform structure of transformers with a pyramid of pooling layers to build models efficient in terms of inference speed. (Mehta and Rastegari, 2021; Maaz *et al.*, 2022; Yang *et al.*, 2022) combine the strength of CNNs and VITs to build efficient models. (Mehta and Rastegari, 2021) combines transformers with MobileNetV2 (Sandler *et al.*, 2018) to obtain global representations. (Maaz *et al.*, 2022) splits the input into multiple channels and performs depth-wise convolution and self-attention across channels to increase the receptive field. (Yang *et al.*, 2022) introduce local self-attention into the convolution within the kernel to capture low-level features. Different from these methods (Huang *et al.*, 2022), proposed convolution-free transformers. They introduced learnable tokens to capture global dependencies. (Zhang *et al.*, 2022) perform weight multiplexing across consecutive transformer blocks, i.e., they share weights across layers while imposing a transformation on the weights to increase diversity. In this work, similar to (Zhang *et al.*, 2022), we share weights. However, in AN-ViT, weights are shared between AN-Large and AN-Small for each block. Furthermore, we train both the large and small models together.

Chapter 3

ADJOINED VIT

We use Adjoined training paradigm to compress vision transformers (ViTs). Before understanding Adjoined ViT (AN-ViT), let us have a re-look at ViT. ViT model first divides the input image into patches that are further tokenized to embedding dimension E. A class token is added to the image token to form an input $x \in R^{N \times E}$, where N represents the total number of tokens. The input tokens are passed through a series of transformer blocks. Each transformer block consists of a multi-head self attention (MSA) and a multi-layer perceptron (MLP) module. We use the class token from the last block to make the final classification.

The MSA module linearly transforms the input tokens x into queries(Q), keys(K), and values(V). Now, instead of performing single self-attention (Vaswani *et al.*, 2017), the MSA module linearly projects the queries, keys, and values H number of times with different learned linear projections. Each group of Q, K, and V is referred to as a head. Next, each head performs a self-attention operation. The output from each head is concatenated and projected back to E dimension (3.1). The output from the MSA module is further passed through the MLP module. We calculate self-attention using the following equations.

$$\begin{aligned} &\text{MSA}(Q, K, V) = \text{Concat}(head_1, head_2, ...head_H)W^O, \\ &\text{where } head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \\ &W_i^Q \in R^{E \times d_Q}, W_i^K \in R^{E \times d_K}, W_i^V \in R^{E \times d_V}, W^O \in R^{Hd_V \times E} \end{aligned} \tag{3.1}$$

In adjoined training paradigm, each MSA module receives two inputs $x_{large} \in R^{N \times E}$ and $x_{small} \in R^{N \times \frac{E}{\alpha}}$. The vector x_{large} represents an input to the original (large)

network, while the vector x_{small} is the input to the smaller (compressed network). Consequently, the MSA block produces two outputs MSA_{large} and MSA_{small}, corresponding to the large and small models. Each transformer block in the AN paradigm is associated with compression factor (α). The compression factor decides the number of heads in the smaller model. We calculate self-attention for both models using the following equations.

$$\begin{aligned}
&\text{MSA}_{large}(Q_{large}, K_{large}, V_{large}) = \text{Concat}(head_1, head_2, ...head_H)W^{O_{large}}, \\
&\text{where } head_i = \text{Attention}(Q_{large}W_i^{Q_{large}}, K_{large}W_i^{K_{large}}, V_{large}W_i^{V_{large}}), \\
&W_i^{Q_{large}} \in R^{E\times d_Q}, W_i^{K_{large}} \in R^{E\times d_K}, W_i^{V_{large}} \in R^{E\times d_V}, W^{O_{large}} \in R^{Hd_V\times E}
\end{aligned} \tag{3.2}$$

$$\begin{aligned}
&\text{MSA}_{small}(Q_{small}, K_{small}, V_{small}) = \text{Concat}(head_1, head_2, ...head_{\frac{H}{\alpha}})W^{O_{small}}, \\
&\text{where } head_i = \text{Attention}(Q_{small}W_i^{Q_{small}}, K_{small}W_i^{K_{small}}, V_{small}W_i^{V_{small}}), \\
&W_i^{Q_{small}} \in R^{\frac{E}{\alpha}\times d_Q}, W_i^{K_{small}} \in R^{\frac{E}{\alpha}\times d_K}, W_i^{V_{small}} \in R^{\frac{E}{\alpha}\times d_V}, W^{O_{small}} \in R^{\frac{H}{\alpha}d_V\times \frac{E}{\alpha}}
\end{aligned} \tag{3.3}$$

As shown in Figure 3.1, all parameters of MSA_{small} are shared with MSA_{large}. MSA_{large} is similar to the MSA module in standard ViTs. However, there are two important distinctions in MSA_{small} that leads to network compression. First, each group of Q, K, and V is linearly projected H/α number of times instead of H times. Second, the concatenation of all heads is projected to the E/α dimension. Thus the output dimensions of MSA_{large} and MSA_{small} modules are $R^{N\times E}, R^{N\times \frac{E}{\alpha}}$ respectively. For the first transformer block $x_{large} = x_{small}$, but the two vectors are not necessarily equal for the deeper transformer blocks (Figure 3.1). The output from the MSA_{large} and MSA_{small} modules are further passed to MLP_{large} and MLP_{small} respectively. MLP_{large} is the same as in standard ViT. However, we decrease the input and output channels of all linear layers within MLP_{small} by a factor of α.

Putting all this together, we can compress any transformer-based model using the Adjoined training paradigm. Since the first block receives a single input (Figure 3.1),

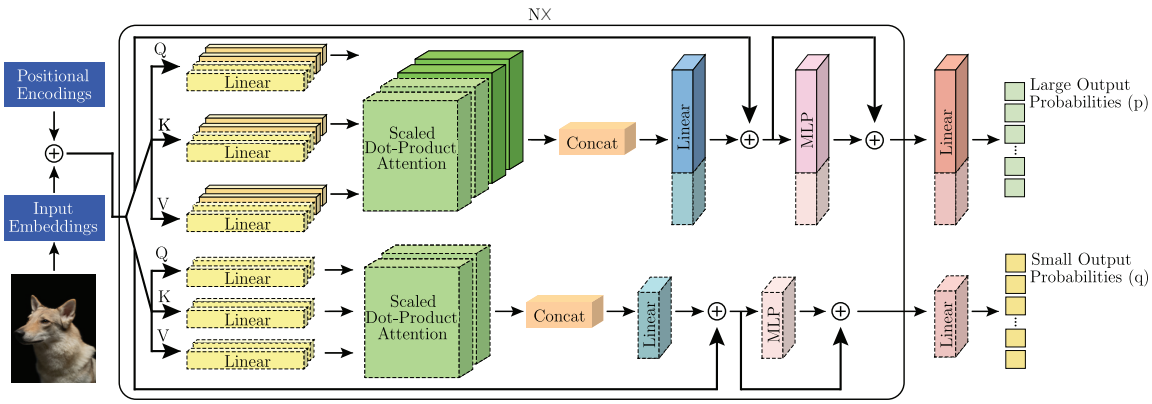

Figure 3.1: Training Paradigm Based on Adjoined Vision Transformer (AN-ViT). The Original and the Compressed Version of the Network are Trained Together with the Parameters of the Smaller Network Shared Across Both. The Network Outputs Two Probability Vectors p (Original Network) and q (Smaller Network).

we create two copies that are passed to the AN-ViT. The network finally gives two output probabilities p, q corresponding to the large and small (compressed) networks. We train the network using adjoined loss function (3.4), which forces p and q to be close to one another.

3.1 Adjoined Loss

Let y be the ground-truth one-hot encoded vector and p and q be output probabilities by the AN-ViT. Then

$$L(y, p, q) \quad = \quad -y \log p \quad + \quad \lambda(t)\, KL(p, q) \tag{3.4}$$

where $KL(p, q) = \sum_i p_i \log \frac{p_i}{q_i}$ is the measure of difference between two probability measures (Kullback and Leibler, 1951). The regularization term $\lambda : [0, 1] \rightarrow R$ is a function which changes with the number of epochs during training.

Here $t = \frac{\text{current epoch}}{\text{Total number of epochs}}$ equals zero at the start of training and equals one at the end.

The first term in the loss function is the cross-entropy loss that trains the larger network. The second term forces the output of the smaller model to be similar to the larger model. We slowly increase the value of the regularization term λ as we want the smaller model to slowly start learning from the pre-trained larger model instead of learning everything in one go. In our experiments, we used $\lambda(t) = \min\{4t^2, 1\}$. Thus, the KL term is initially zero and steadily grows to one at 50% training.

3.2 Differentiable Adjoined ViT

In AN-ViT, we choose a fixed compression factor (α) for all blocks. However, different blocks capture different features and thus may be compressed to different compression ratios. Choosing α independently for each block would add more flexibility and possibly improve the performance of the current framework. Thus, we propose Differentiable Adjoined-ViT(DAN-ViT).

In the DAN architecture, now we are using $\alpha \in A = \{\alpha_1, ..., \alpha_n\}$, where α_x is a factor of the number of heads in the MSA of the encoder block. To find the optimal network structure, we had to solve $\arg\max_{\alpha \in A} L(q)$, where q is the output probability vector and L is the loss function. This problem can be solved for one encoder block by computing $L(q)$ first and then the maximum. But the complexity of this problem increases with the increase in encoder blocks. So, for a L-layer architecture, the search space is n^L, where n is the number of α's.

To solve this issue, we are using Gumbel-softmax, a re-parametrization trick that can be viewed as a differentiable approximation to the argmax function. Now, the problem can be formulated as $\sum_{\alpha \in A} g_\alpha L(q_\alpha)$, where g_α are the gumbel weights corresponding to the particular α. It is now differentiable and can easily be solved using back-propagation.

Now, we will replace the standard encoder blocks with the DAN encoder blocks.

The input and output dimensions, and the output z_1 remain the same as described in the AN architecture. The output z_2 of the DAN encoder block is defined below:

$$z_i^{'} = MSA_{DAN}(LN(z_{02})) + z_{02}$$

$$z^{'} = \sum_{i=1}^{m} g(\eta)_i z_i^{'}$$

$$z_i = MLP_{DAN}(LN(z^{'})) + z^{'}$$

$$z_2 = \sum_{i=1}^{m} g(\eta)_i z_i \tag{3.5}$$

where $\eta = [\eta_1, ..., \eta_m]$ denotes the mixing weights, and $z_i^{'}$ and z_i are the outputs for MSA and MLP respectively, corresponding to the different α's and g is the gumbel-softmax function.

Given a vector $v = [v_1, ..., v_n]$ and a constant τ, the gumbel-softmax (Wan *et al.*, 2020) function is defined as $g(v) = [g_1, ..., g_n]$, where:

$$g_i = \frac{\exp[(v_i + \epsilon_i)/\tau]}{\sum_i \exp[(v_i + \epsilon_i)/\tau]} \tag{3.6}$$

and $\epsilon_i \in N(0,\ 1)$ is uniform random noise, or "gumbel noise."

3.3 Differentiable Adjoined Loss

Let the search space be $A = \{\alpha_1, ..., \alpha_m\}$ Let y be the one-hot-encoded vector of gold labels, and p and q be the output probabilities of the Differentiable Adjoined Network. Then

$$L(y, p, q) = -y \log p + \lambda(t)(KL(p, q) + \gamma n_f(H)) \tag{3.7}$$

where $KL(p, q)$, $\lambda(t)$ are the same as used in Adjoined loss, and $H = [\eta_1, ..., \eta_l]$, where η_i is the mixing weight vector for the i^{th} encoder block. n_f represents the gumbel

weighted floating point operations for the given network. That is,

$$n_f(H) = \sum_{\eta_i \in H} \sum_{j=1}^{m} g(\eta_i)_j FLOPs(i, \alpha_j) \tag{3.8}$$

where $FLOPs(i, \alpha_j)$ measures the number of floating point operations at the i^{th} encoder block corresponding to the hyper-parameter α_j; γ is a constant which normalizes the n_f term as the number of FLOPs corresponding to any setting of the mixing weights can be very large.

The major distinction between Differentiable Adjoined Loss and Adjoined Loss is the n_f term. As large networks tend to have more accuracy, DAN will favor networks with low α or large networks. So, n_f term serves as a regularization penalty against DAN's preference for large networks. After training DAN-ViT, we choose the compression factor corresponding to the maximum Gumbel weight in each block. The searched architecture is then trained in Adjoined fashion using the adjoined loss function (3.4).

3.4 Chapter Summary

Adjoined Network(AN) training forces the compressed network to generate output probabilities that are comparable to those of the original or large network. This allows us to prune sparse tensors, which are computationally expensive and necessitate specialized operations to handle matrix sparsity. DAN-ViT works similarly to AN-ViT, with the exception that instead of keeping the compression factor α constant across all blocks, we can choose independent compression factor values. This increases network flexibility and improves network performance. With fewer computations and parameters, inference times are drastically reduced. Additionally, we can achieve comparable levels of precision in the compressed network as well the large network.

Chapter 4

EXPERIMENTS

In this section, we conduct experiments on the ImageNet (Russakovsky *et al.*, 2015) dataset to show the effectiveness of Adjoined training paradigm for the compression of vision transformers. We conduct our experiments with two ViT variants (1) T2T-ViT (Yuan *et al.*, 2021) and (2) Swin-Transformers (Liu *et al.*, 2021). For both models, we replace their transformer block with Adjoined transformer block (as discussed in Chapter 3). AN-ViTs are then trained in adjoined fashion using the Adjoined loss function (3.4). In the AN paradigm, we obtain two networks. In this section, we refer to them by X-AN-Large-α and X-AN-Small-α, where X represents the model type, and α represents the compression factor as defined in Chapter 3. For example, Swin-AN-Small-2 represents the compressed Swin Transformer trained via the AN paradigm with $\alpha = 2$, and Swin-AN-Large-2 represents the original (full) Swin Transformer trained via the AN paradigm with $\alpha = 2$. Furthermore, we conduct experiments with DAN-ViT. DAN-ViT searches for the optimal α value within each transformer block. Similar to AN, we refer to them as X-DAN-Large and X-DAN-Small. The search space for α values is set to $(1, 2, 3)$ and $(1, 2, 4, 8)$ for T2T-DAN and Swin-DAN respectively.

We ran our experiments on GPU enabled machine using Pytorch. For both models (T2T-ViT and Swin Transformer), we use the same data augmentations and hyper-parameters used by the original authors. We train both ImageNet pre-trained models in Adjoined fashion for 50 epochs with a constant learning rate. The learning rate and batch size is set to 10^{-3}, 128 for T2T-ViT-AN and 10^{-5}, 64 for Swin-AN.

For DAN-ViT, we conduct the searching stage on the ImageNet-100 dataset (Tian

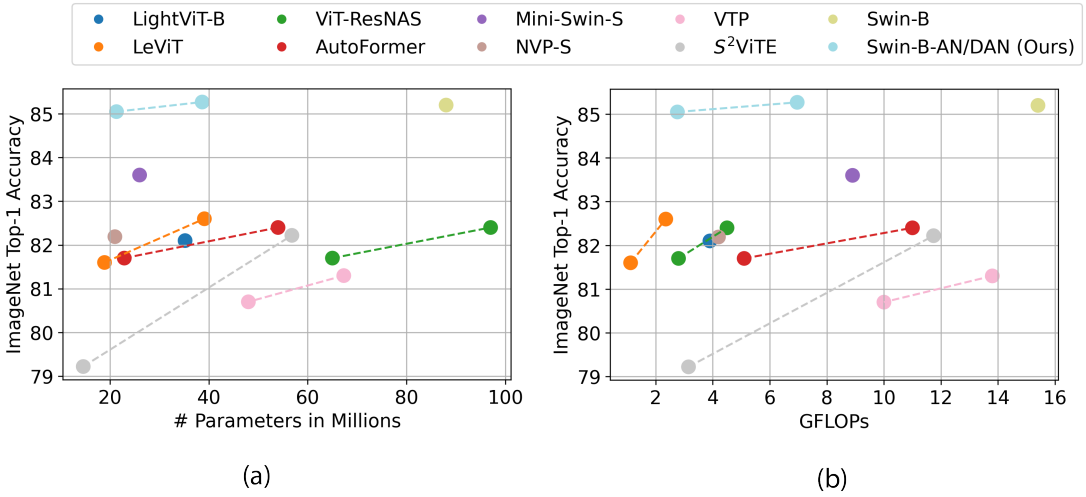

Figure 4.1: The Comparison Between AN-ViT/DAN-ViT and Transformer Based Models Such as Swin Transformers (Liu *et al.*, 2021) and Mini-Swin (Zhang *et al.*, 2022) on the ImageNet Dataset Against (a) Number of Parameters, and (b) FLOPs. Our Models Significantly Outperform All Other ViTs in Terms of Top-1 Accuracy on the ImageNet Dataset While Being Smaller and Faster.

et al., 2020a). ImageNet-100 is a subset of the ImageNet-1k dataset with 100 classes and about $130k$ images. We train both DAN-ViT models for 100 epochs with the same hyper-parameters as in their corresponding AN-ViT models. The γ in (**??**) is set to 10^{-12} for both models. The model searched using DAN-ViT is then trained in adjoined fashion using AN-ViT.

4.1 Comparison against SOTA efficient ViTs

In this section, we compare ViTs compressed by Adjoined Network against other current SOTA-efficient ViTs. In Table 4.1, we observe that models compressed by AN/DAN achieve significantly higher accuracy than other similar-sized models on the ImageNet dataset. Swin-B-AN achieves approximately $\sim (3.5\% - 1.5\%)$ higher accuracy than similar sized model such as LeViT (Huang *et al.*, 2022), AutoFormer-small (Chen *et al.*, 2021a) and Mini-Swin-S (Zhang *et al.*, 2022). Compared to other efficient models, Swin-B-AN exhibits one of the lowest FLOPs. Swin-B-AN surpasses the model with next highest accuracy (Mini-Swin-B) by 0.75% while being 5.7×

Model	Accuracy	# Params (M)	GFLOPs
LightViT-B (Huang *et al.*, 2022)	82.1	35.2	3.9
LeViT-256 (Graham *et al.*, 2021)	81.6	18.9	1.12
LeViT-384 (Graham *et al.*, 2021)	82.6	39.1	2.353
ViT-ResNAS-Small (Liao *et al.*, 2021)	81.7	65	2.8
ViT-ResNAS-Medium (Liao *et al.*, 2021)	82.4	97	4.5
AutoFormer-small (Chen *et al.*, 2021a)	81.7	22.9	5.1
AutoFormer-base (Chen *et al.*, 2021a)	82.4	54	11
Mini-Swin-S (Zhang *et al.*, 2022)	83.6	26	8.9
Mini-Swin-B (Zhang *et al.*, 2022)	84.3	46	15.7
NVP-S (Yang *et al.*, 2021)	82.2	21	4.2
T2T-UVC (Yu *et al.*, 2022)	79.6	-	2.47
VTP (20% pruned) (Zhu *et al.*, 2021)	81.3	67.3	13.8
VTP (40% pruned) (Zhu *et al.*, 2021)	80.7	48.0	10.0
S^2ViTE-Small (Chen *et al.*, 2021b)	79.22	14.6	3.15
S^2ViTE-Base (Chen *et al.*, 2021b)	82.22	56.8	11.74
Swin-B-AN-Small-4 (Our)	<u>85.05</u>	21.32	2.76
Swin-B-DAN (Our)	**85.27**	38.66	6.96

Table 4.1: The Table Compares the Performance of AN-ViT and DAN-ViT Against SOTA Efficient ViTs on ImageNet Dataset. Accuracy Represents the Top-1 Accuracy on ImageNet Dataset. # Params(M), GFLOPs Represents the Number of Parameters (in Millions) and GFLOPs of the Model Respectively.

faster and 2.1× smaller. DAN-ViT further increases the accuracy. Amongst the compared models, Swin-B-DAN achieves the highest top-1 accuracy of 85.27% on the ImageNet dataset. We observe similar results in Figure 4.1. Models compressed with our training paradigm are explicitly on the top left of the graph, while other methods are clustered on the lower half.

Model	Accuracy	# Params (M)	GFLOPs	Params ↓	GFLOPs ↓
Swin-B (Liu *et al.*, 2021)	85.2	88	15.4	1X	1X
Swin-B-AN-Small-4 (Our)	85.05	21.32	2.76	4.13X	5.58X
Swin-B-DAN (Our)	85.27	38.66	6.96	2.28X	2.21X
T2T-ViT (Yuan *et al.*, 2021)	81.5	21.5	4.8	1X	1X
T2T-ViT-AN-Small-2 (Our)	80.4	10.2	2.16	2.11X	2.22X
T2T-ViT-DAN (Our)	81.3	16.5	3.36	1.31X	1.33X

Table 4.2: The Table Evaluates the Performance of AN-ViT and DAN-ViT Against the Base/Standard Model on ImageNet Dataset. Accuracy, # Params and GFLOPs Are Same as in Table 4.1. Param ↓, GFLOPs ↓ Represents the Ratio of Parameters and GFLOPs Compared to the Base Model.

4.2 Compression

In this section, we evaluate the performance of ViTs compressed by AN and DAN. Table 4.2, compares the performance of ViTs compressed using AN and DAN against the standard model. AN-ViT successfully compresses both transformer models (Swin Transformers and T2T-ViT) without any significant loss in top-1 accuracy on the ImageNet dataset. Swin-B-AN-Small-4 while being 4.1× smaller and 5.58× faster suffers only 0.15% drop in accuracy. Furthermore, ViTs trained via DAN searches for the optimal compression factor for each block and thus may even surpass the base model's performance. Swin-B-DAN exceeds the base model by 0.07% while achieving a 2.2× reduction in model size and 2.1× reduction in FLOPs. We observe similar results for the T2T-ViT model. We also observe that Swin-B is the larger model, thus, can be compressed more without any significant loss in accuracy.

Chapter 5

CONCLUSIONS

This chapter presents a summary of this thesis work and future research scopes in the field of compression of any tranformer-based architectures.

5.1 Summary

In this thesis work, we proposed Adjoined Training paradigm for the compression of any transformer-based architecture. AN-ViT trains the compressed and the base model together, wherein all the weights of the compressed model are shared with the base model. AN-ViT compresses Swin-Transformers by 4.1× and 5.5× in the number of parameters and FLOPs, respectively, while achieving 85.05% top-1 ImageNet accuracy (i.e., 0.15% loss in accuracy). We further propose Differentiable Adjoined Vision Transformer (DAN-ViT) that searches for the optimal compresses factor for each block of AN-ViT. Augmenting AN-ViT with DAN-ViT enhances flexibility and improves the performance of the compressed model. Swin-Transformer compressed by DAN-ViT exceeds base network in term top-1 accuracy by 0.07% and achieves 85.27% ImageNet accuracy while exhibiting 2.2× fewer parameters and FLOPs.

5.2 Future Research

In this work, we focus on the image classification task. In the future, we plan to experiment with AN-ViT for other computer vision tasks, such as object detection and segmentation. It would also be interesting to see the performance of adjoined training paradigm for the compression of transformers used in NLP tasks.

REFERENCES

Ahn, S., S. X. Hu, A. Damianou, N. D. Lawrence and Z. Dai, "Variational information distillation for knowledge transfer", arXiv preprint arXiv:1904.05835 (2019).

Cai, H., L. Zhu and S. Han, "Proxylessnas: Direct neural architecture search on target task and hardware", in "7th International Conference on Learning Representations, ICLR", (2019).

Chen, M., H. Peng, J. Fu and H. Ling, "Autoformer: Searching transformers for visual recognition", in "Proceedings of the IEEE/CVF international conference on computer vision", pp. 12270–12280 (2021a).

Chen, T., Y. Cheng, Z. Gan, L. Yuan, L. Zhang and Z. Wang, "Chasing sparsity in vision transformers: An end-to-end exploration", Advances in Neural Information Processing Systems **34**, 19974–19988 (2021b).

Chu, X., Z. Tian, B. Zhang, X. Wang, X. Wei, H. Xia and C. Shen, "Conditional positional encodings for vision transformers", arXiv preprint arXiv:2102.10882 (2021).

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale", ICLR (2021).

Evci, U., T. Gale, J. Menick, P. S. Castro and E. Elsen, "Rigging the lottery: Making all tickets winners", arXiv preprint arXiv:1911.11134 (2019).

Gale, T., E. Elsen and S. Hooker, "The state of sparsity in deep neural networks", arXiv preprint arXiv:1902.09574 (2019).

Graham, B., A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou and M. Douze, "Levit: a vision transformer in convnet's clothing for faster inference", in "Proceedings of the IEEE/CVF international conference on computer vision", pp. 12259–12269 (2021).

Han, K., A. Xiao, E. Wu, J. Guo, C. Xu and Y. Wang, "Transformer in transformer", Advances in Neural Information Processing Systems **34**, 15908–15919 (2021).

Han, S., H. Mao and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding", arXiv preprint arXiv:1510.00149 (2015).

Hinton, G. E., O. Vinyals and J. Dean, "Distilling the knowledge in a neural network", CoRR **abs/1503.02531** (2015).

Huang, T., L. Huang, S. You, F. Wang, C. Qian and C. Xu, "Lightvit: Towards light-weight convolution-free vision transformers", arXiv preprint arXiv:2207.05557 (2022).

Kim, J., S. Park and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer", arXiv preprint arXiv:1802.04977 (2018).

Krizhevsky, A., I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", Communications of the ACM **60**, 6, 84–90 (2017).

Kullback, S. and R. A. Leibler, "On information and sufficiency", The annals of mathematical statistics **22**, 1, 79–86 (1951).

Kusupati, A., V. Ramanujan, R. Somani, M. Wortsman, P. Jain, S. Kakade and A. Farhadi, "Soft threshold weight reparameterization for learnable sparsity", arXiv preprint arXiv:2002.03231 (2020).

Li, H.-T., S.-C. Lin, C.-Y. Chen and C.-K. Chiang, "Layer-level knowledge distillation for deep neural network learning", Applied Sciences **9**, 10 (2019).

Li, Y., H. Mao, R. Girshick and K. He, "Exploring plain vision transformer backbones for object detection", in "Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX", pp. 280–296 (Springer, 2022).

Liao, Y.-L., S. Karaman and V. Sze, "Searching for efficient multi-stage vision transformers", arXiv preprint arXiv:2109.00642 (2021).

Lin, M., R. Ji, Y. Wang, Y. Zhang, B. Zhang, Y. Tian and L. Shao, "Hrank: Filter pruning using high-rank feature map", in "Proceedings of the IEEE/CVF conference on computer vision and pattern recognition", pp. 1529–1538 (2020a).

Lin, M., R. Ji, Y. Zhang, B. Zhang, Y. Wu and Y. Tian, "Channel pruning via automatic structure search", arXiv preprint arXiv:2001.08565 (2020b).

Liu, H., K. Simonyan and Y. Yang, "Darts: Differentiable architecture search", in "7th International Conference on Learning Representations, ICLR", (2019).

Liu, Z., Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows", in "Proceedings of the IEEE/CVF international conference on computer vision", pp. 10012–10022 (2021).

Maaz, M., A. Shaker, H. Cholakkal, S. Khan, S. W. Zamir, R. M. Anwer and F. S. Khan, "Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications", arXiv preprint arXiv:2206.10589 (2022).

Mehta, S. and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer", arXiv preprint arXiv:2110.02178 (2021).

Park, S. T., D. Kim and N. Kwak, "Relational knowledge distillation", arXiv preprint arXiv:1904.05068 (2019).

Peng, H., R. Jin, Z. Liu, J. Zhang, L. Wang and T. Zhang, "Correlation congruence for knowledge distillation", arXiv preprint arXiv:1904.01802 (2019).

Real, E., S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le and A. Kurakin, "Large-scale evolution of image classifiers", in "Proceedings of the 34th International Conference on Machine Learning, ICML", edited by D. Precup and Y. W. Teh, vol. 70 of *Proceedings of Machine Learning Research*, pp. 2902–2911 (PMLR, 2017).

Romero, A., N. Ballas, S. E. Kahou, A. Chassang, C. Gatta and Y. Bengio, "Fitnets: Hints for thin deep nets", in "3rd International Conference on Learning Representations, ICLR", (2015).

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge", International journal of computer vision **115**, 3, 211–252 (2015).

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 4510–4520 (2018).

Strudel, R., R. Garcia, I. Laptev and C. Schmid, "Segmenter: Transformer for semantic segmentation", in "Proceedings of the IEEE/CVF international conference on computer vision", pp. 7262–7272 (2021).

Tan, M., B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile", in "IEEE Conference on Computer Vision and Pattern Recognition, CVPR", pp. 2820–2828 (2019).

Tian, Y., D. Krishnan and P. Isola, "Contrastive multiview coding", in "European conference on computer vision", pp. 776–794 (Springer, 2020a).

Tian, Y., D. Krishnan and P. Isola, "Contrastive representation distillation", in "8th International Conference on Learning Representations, ICLR", (2020b).

Touvron, H., M. Cord, M. Douze, F. Massa, A. Sablayrolles and H. Jégou, "Training data-efficient image transformers & distillation through attention", in "International conference on machine learning", pp. 10347–10357 (PMLR, 2021).

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need", Advances in neural information processing systems **30** (2017).

Wan, A., X. Dai, P. Zhang, Z. He, Y. Tian, S. Xie, B. Wu, M. Yu, T. Xu, K. Chen *et al.*, "Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition", pp. 12965–12974 (2020).

Wang, W., E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions", in "Proceedings of the IEEE/CVF international conference on computer vision", pp. 568–578 (2021).

Wu, B., X. Dai, P. Zhang, Y. Wang, F. Sun, Y. Wu, Y. Tian, P. Vajda, Y. Jia and K. Keutzer, "Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search", in "IEEE Conference on Computer Vision and Pattern Recognition, CVPR", pp. 10734–10742 (2019).

Wu, H., B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan and L. Zhang, "Cvt: Introducing convolutions to vision transformers", in "Proceedings of the IEEE/CVF International Conference on Computer Vision", pp. 22–31 (2021).

Xie, E., W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers", Advances in Neural Information Processing Systems **34**, 12077–12090 (2021).

Yang, C., Y. Wang, J. Zhang, H. Zhang, Z. Wei, Z. Lin and A. Yuille, "Lite vision transformer with enhanced self-attention", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition", pp. 11998–12008 (2022).

Yang, H., H. Yin, P. Molchanov, H. Li and J. Kautz, "Nvit: Vision transformer compression and parameter redistribution", arXiv preprint arXiv:2110.04869 (2021).

Yim, J., D. Joo, J. Bae and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning", in "IEEE Conference on Computer Vision and Pattern Recognition, CVPR", pp. 7130–7138 (2017).

Yu, S., T. Chen, J. Shen, H. Yuan, J. Tan, S. Yang, J. Liu and Z. Wang, "Unified visual transformer compression", arXiv preprint arXiv:2203.08243 (2022).

Yuan, L., Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet", in "Proceedings of the IEEE/CVF international conference on computer vision", pp. 558–567 (2021).

Zagoruyko, S. and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer", in "5th International Conference on Learning Representations, ICLR", (2017).

Zhang, C., W. Li, W. Ouyang and D. Xu, "Deep mutual learning", arXiv preprint arXiv:1706.00384 (2017).

Zhang, J., H. Peng, K. Wu, M. Liu, B. Xiao, J. Fu and L. Yuan, "Minivit: Compressing vision transformers with weight multiplexing", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition", pp. 12145–12154 (2022).

Zhu, M. and S. Gupta, "To prune, or not to prune: exploring the efficacy of pruning for model compression", arXiv preprint arXiv:1710.01878 (2017).

Zhu, M., Y. Tang and K. Han, "Vision transformer pruning", arXiv preprint arXiv:2104.08500 (2021).

Zoph, B. and Q. V. Le, "Neural architecture search with reinforcement learning", in "5th International Conference on Learning Representations, ICLR", (2017).