Project 1

Group 2

8/09/2023

1 Introduction

In the field of deep learning (DL), it is a common practice to employ large scale models that have been pre-trained on extensive and diverse datasets from the general domain [3]. These pre-trained models are then re-adapted to perform specific downstream tasks, avoid training a new model from scratch [7], through a process also known as *fine-tuning*. Due to its ease and effectiveness, this paradigm has already been used to deploy large, fine-tuned models across a variety of real-world applications [6]. Traditionally, this fine-tuning technique involves updating all the parameters of the pre-trained model using domain-specific data. This alone results in significant improvements, as it eliminates the need for a substantial portion of the training process, thereby conserving valuable time and computational resources. However, as deep models like Transformers [8] have evolved and advanced, this traditional approach has become excessively resource-intensive in terms of storage and computational demands. To address this challenge, innovative techniques have emerged, with a focus on adapting only specific parameters or incorporating external modules for new tasks. In this context, adaptation has proven to be a crucial aspect of fine-tuning in deep learning, particularly in the field of natural language processing, allowing a single large-scale, pre-trained language model to be tailored for various downstream applications [1, 4].

Low-Rank Adaptation (LoRA), introduced in [5], offers an efficient alternative to the conventional fine-tuning process. LoRA allows us to fine-tune pre-trained models without the need to directly update all of their parameters. This technique is particularly valuable when dealing with large state-of-the-art models, significantly reducing the computational and memory demands associated with fine-tuning.

In this project, our primary objective is to explore and compare two fine-tuning strategies: full fine-tuning and LoRA, using a pre-trained Vision Transformer (ViT) model [5]. Our experimentation will be centered on ImageWoof, a dataset comprising images of ten distinct dog breeds, derived from the broader ImageNet dataset. The ultimate goal of our fine-tuning endeavor is to train a model capable of accurately classifying these ten dog breeds.

2 Methods

2.1 Dataset

For this project, we employed a subset of ImageWoof, a derivative of the ImageNet dataset [2]. The ImageWoof dataset consists of a diverse collection of dog images from ten different classes. An overview visualization can be found in Figure 1.

2.2 Pre-trained Vision Transformer (ViT)

2.2.1 Model Selection

The ViT model used in our study is the '*vittinypatch*16224,' which serves as the backbone architecture for our image classification task. This specific model, *vit tiny patch16 224*, is an instance of ViT-Tiny that has undergone pretraining on the ImageNet-21k dataset and further fine-tuning on the ImageNet2012 dataset. It is designed to handle input images with a resolution of 224x224 pixels, and it divides the input into patches of size 16x16 pixels for processing. As part of the pre-processing pipeline, we resized the original images from the ImageWoof dataset to conform to the specified dimensions and applied normalization.

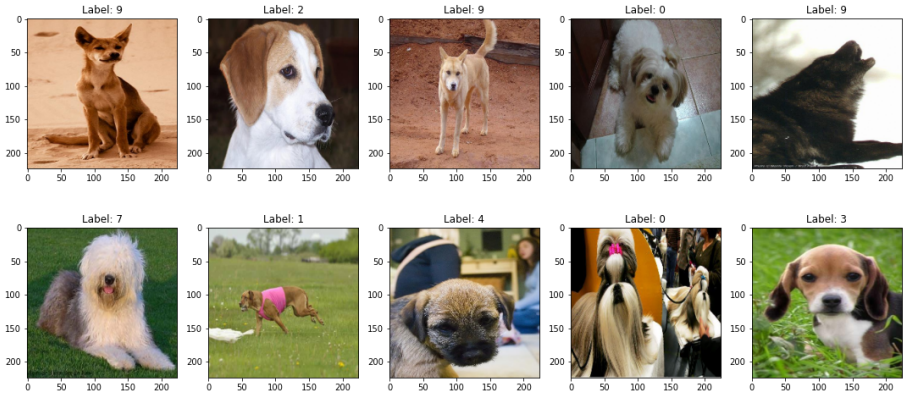

Figure 1: ImageWoof dataset: different istances of images from the different classes.

2.3 Fine-tuning Procedures

To leverage the pretraining on a large-scale dataset, we initialized the ViT model with pretrained weights from the original ImageNet dataset. The full fine-tuning process was conducted as follow. To initiate the training process, we replaced the classification head of the original model with a new one, precisely tailored to register the correct number of categories in our dataset. The training was conducted using the Adam optimizer with a batch size of 128 over a span of five epochs. The learning rate was set to 0.0004, and the loss function employed was Cross Entropy. Throughout the training process, we monitored the total training time, as well as the accuracy achieved at the end of each epoch. In addition to training, we also assessed the model's performance on the testing set to evaluate its generalization and predictive capabilities.

Subsequently, we fine-tuned the ViT by using LoRA. LoRA employs an innovative approach to training dense layers within a neural network while preserving the original pre-trained weights in a frozen state. Instead of directly optimizing the weights themselves during adaptation or fine-tuning, LoRA focuses on the rank decomposition matrices that capture the changes in weights. To be more precise, when adapting a pre-trained weight matrix, denoted as W_0, the LoRA-adapted model's output, x, can be expressed as:

$$h = (W_0 + \Delta W)x = W_0 x + BAx,$$

where $W_0 \in \mathbb{R}^{dxk}$, $B \in \mathbb{R}^{dxr}$, $A \in \mathbb{R}^{rxk}$, and $r = min(d, k)$ is the rank.

To apply LoRA to the dense layers of the ViT, we designed our own wrapper class, which initializes new dense layers for the LoRA matrices.

3 Results

We first present the standard fine-tuning procedure results on ImageWoof. We then show the results obtained introducing the adaptation. Finally, we visualize the final attention maps of the ViT, together with a proper comparison of attention changes from the original ViT to the ViT and the LoRA model.

3.1 Full Fine-tuning

The ViT is designed to take images as inputs, prepare the training data, construct and train the model, ensuring that it classifies them accurately into their respective classes. The data is divided into training and test sets to assess the model's performance on the test set. Throughout the training process, the model's current state is assessed on the training set at each step of the algorithm. Additionally, a hold-out validation is conducted on a dataset separate from the training data.

Figure 2 displays the learning curves for the problem at hand. Here, we illustrate the progress related to learning metrics during training. Initially, we observe a high training loss, which gradually decreases as more training examples are added, eventually plateauing near zero. On the left-hand side, the accuracy plot shows a consistent upward trend with increasing training epochs. This signifies that the model's performance steadily improves as it gains more exposure to the training data. Following

an initial learning phase, the accuracy stabilizes and approaches a plateau, indicating that further training may have limited impact on model performance. The final achieved accuracy stands at 79%, demonstrating the model's effective classification of target classes in the dataset. Training took place on the Educloud server, with a total training time of 2263.59 seconds.

Subsequently, the model underwent evaluation on the test set, resulting in a test loss of 0.0023, with a recorded accuracy of 91.65%.

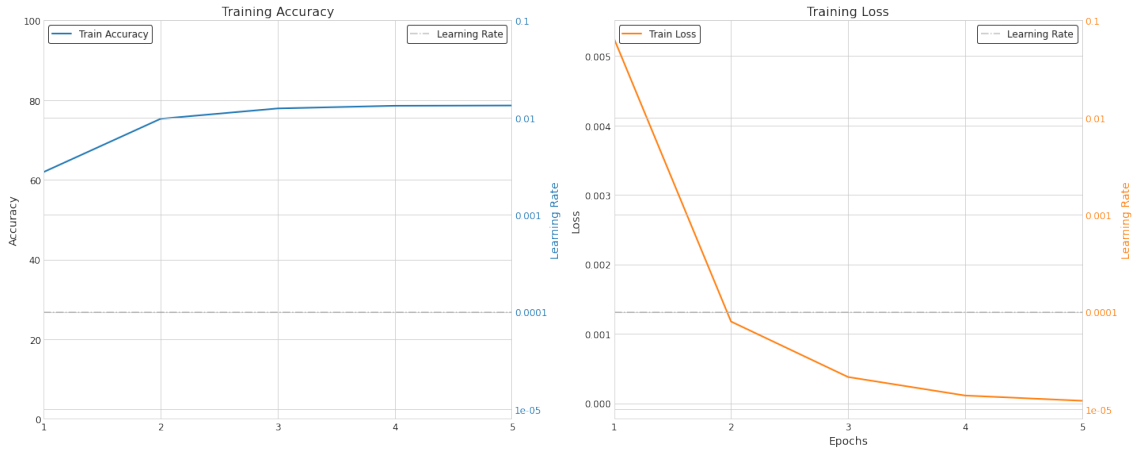

Figure 2: Training accuracy (left) and loss (right) over epochs during model fine-tuning. The left subplot shows the trend of training accuracy as it improves with each epoch, indicating the model's learning progress. Learning rate is also reported on the side. On the right, the loss decreases, signifying the optimization process.

3.2 Fine-tuning & adaptation with LoRA

In the second part of our project, we applied LoRA to fine-tune the ViT model. We visualized the accuracy and training loss during this process in Figure 3. The model was trained for 2501.08 seconds, achieving a final training loss of 0.0025 and accuracy of 71%. Concerning the test phase, we obtained a test loss of 0.0027 and accuracy of 88.78%.

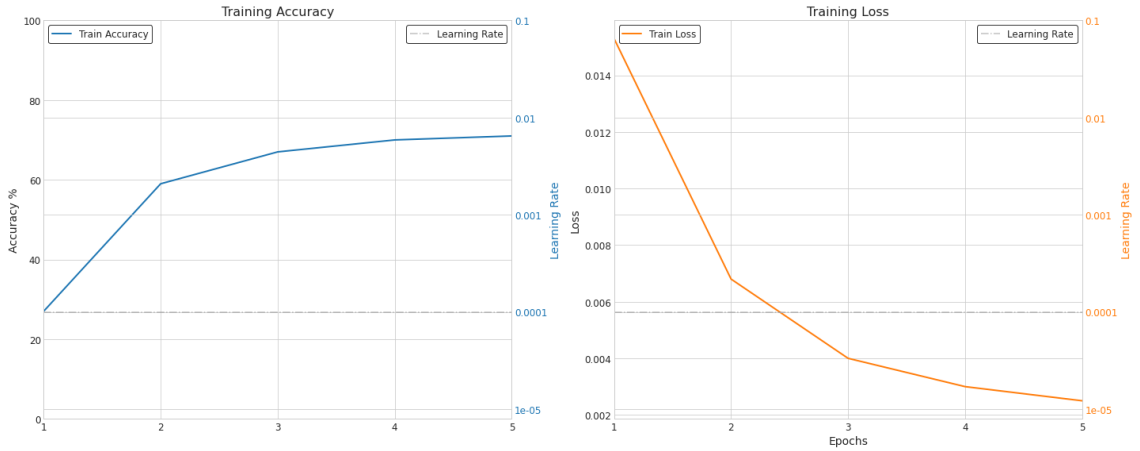

Figure 3: Training accuracy (left) and loss (right) over epochs during model fine-tuning with LoRa adaptation. The left subplot shows the trend of training accuracy as it improves with each epoch, indicating the model's learning progress. Learning rate is also reported on the side. On the right, the loss decreases, signifying the optimization process.

3.3 Visualization of Attention Maps

The visualization of attention maps in the ViT is a crucial component responsible for capturing relationships between different parts of the input image and allowing the model to focus on relevant regions when making predictions. It's similar in concept to the attention mechanism used in natural language processing tasks, like machine translation, but applied to images.

In the ViT, attention maps are calculated using a multi-head self-attention mechanism. This process involves transforming the input image into Query (Q) and Key (K) components through learned linear transformations. The similarity between these components is measured to compute attention scores, highlighting regions of significance in the image. These scores are normalized using a softmax function to derive attention weights, determining the relative importance of image regions. A weighted sum of the Value (V) components, which carry information about each region, is then computed using these attention weights, yielding an attended representation. Multiple attention heads capture different relationships, and positional encodings are added to embed spatial information, enhancing the model's contextual understanding.

3.3.1 Attention Maps for full-finetuned model

Figure 4 shows some attention maps for the full-finetuned model and LoRA model. The model is able to focus exclusively on objects of interest in the image.

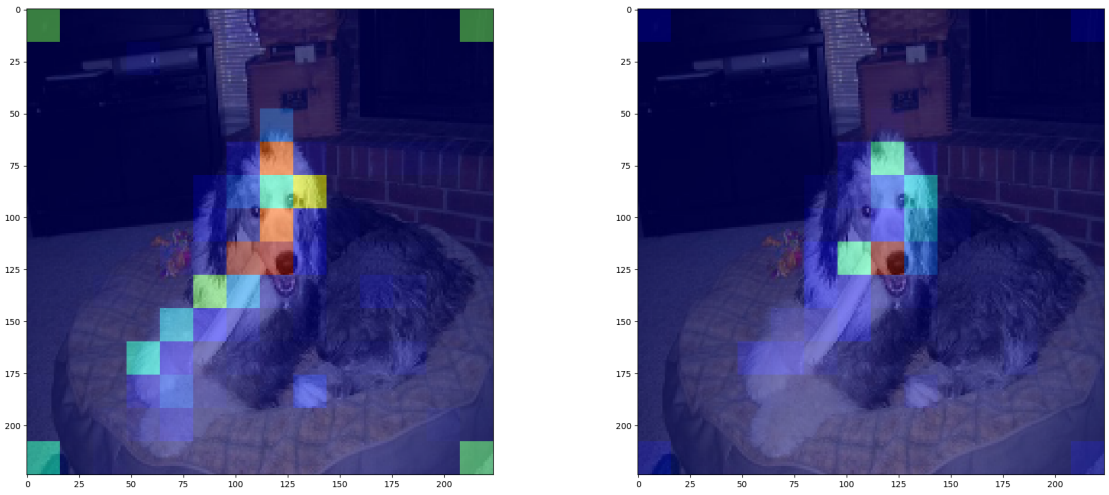

Figure 4: Visualization of attention output of a dog sample: on the left, the attention map for the full-finetuned model. On the right the one obtained for the LoRA model.

4 Discussion

The use of large-scale pre-trained models has become a cornerstone technique in the field of DL. These pre-trained models can be fine-tuned to particular tasks rather than creating new models from scratch. The deployment of large, fine-tuned models across many applications has shown fine-tuning to be extremely effective and efficient. Innovative approaches have been created to address resource-intensive problems the comes with this technique, with a focus on customising some settings or adding external modules for new activities. This adaptation has emerged as a key component of DL optimisation. This allows for the fine-tuning of a single large-scale, pre-trained language model for numerous downstream applications. One such cutting-edge method is LoRA, which was first described in and provides a productive substitute for the traditional fine-tuning procedure by drastically lowering computational and memory requirements.

Using a pre-trained ViT model, our main goal in this study was to investigate and contrast two fine-tuning strategies: complete fine-tuning and LoRA. We then evaluated the flexibility of ViT in the context of fine-tuning and also explore the attention maps. For full-tuning, we got the 78.64% accuracy and loss of 0.00001. The testing accuracy is 91.45% and loss of 0.0023. After adapting with LoRA, we achieved 78.40% accuracy and loss of 0.0001. The testing accuracy is 90.12% and loss of 0.0032. Comparing our model with LoRA to the traditional fine-tuning approach, we observe a significant advantage in terms of parameter efficiency (5526346 vs 177418). The LoRA model demonstrates superior performance while requiring fewer parameters, making it a more resource-efficient and computationally lighter alternative.

In order to understand how the ViT interprets and processes the visual data more thoroughly, we also examined the attention maps that were produced by the ViT during fine-tuning. The model's decision-making process and how it concentrates on relevant data are both key insights revealed by these attention maps. We have been able to generate the attention output for both models. Both models are able to accurately capture the main features that describe the dogs. Taking Figure 4 as reference, we can see that the LoRA model focuses on facial details on the animal. On the contrary the full finetuned model is able to localize other relevant features, such as the legs.

In conclusion, the choice of the model depends on the specific goals of the task. If the primary aim is achieving the highest accuracy, the full fine-tuned model might be the the preferred choice, especially when it benefits from more relevant training data. However, when computational resources are limited, the model with LoRA adaptation offers a compelling alternative. It is designed to adapt to a new tasks efficiently, providing satisfactory results with fewer labeled data requirements.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021.

[4] Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and Luo Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. *arXiv preprint arXiv:2106.03164*, 2021.

[5] Edward J Hu. yelong shen, phillip wallis, zeyuan allen-zhu, yuanzhi li, shean wang, lu wang, and weizhu chen. lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, volume 3, page 7, 2022.

[6] Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pre-trained models. *arXiv preprint arXiv:2004.06660*, 2020.

[7] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, 2020.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.