ADAPTING DINO FOR MEDICAL TASKS USING LORA

tnc197, wxp878, fml137, vxp646, (Class 2, Group 4)

UCPH

ABSTRACT

Foundation models have been a revelation in many machine learning fields, and recently unsupervised foundation models in vision have enjoyed similar success. Now that large foundation models can be used for vision tasks, it is an important question to find the best way to use them. The most common way is by fine-tuning for a specific task which results in large model that is optimized for your task, which does not have to be trained from the ground up. This fine-tuning, however, still requires large amounts of compute as we are still training a foundation model, however briefly. To reduce the computational load of fine-tuning, Low-Rank Adaptation (LoRA) is often used as a way to achieve similar performance by injecting a fractionally insignificant amount of parameters and training only these, with the rest of the model parameters frozen. We show that using LoRA to fine-tune DINOv2 for medical image segmentation tasks can be a way of achieving good task-specific performance with less training time than fine-tuning the entire DINOv2, though these results inconsistent and do not show a significant improvement over training a linear-layer on the frozen features. Our code can be found in the repository: https://github.com/JakobKarrer/ATDL_A3.

Index Terms— LoRa, Foundation Models, DINOv2

1. INTRODUCTION

Our goal is to explore the value of fine-tuning DINOv2 to medical imaging tasks using different modalities. We attempt classification tasks as has already been done in [1] and demonstrate that there is benefit in using LoRA[2] for at least one difficult segmentation task on ultrasound images.

2. BACKGROUND

The major factors behind the success of large language models was parallelization coupled with unsupervised learning. DINOv2[3], a vision foundation model, leverages these two aspects in its training to produce a vision model that achieves state-of-the-art (SOTA) performance on many vision tasks. They achieve the task-specific SOTA performance by simply adding a head on top of the features produced by the frozen DINOv2. To maximize the performance of this method one can fine-tune the entire DINOv2 for the specific task. In the original DINOv2 paper they do this by adding the head on top of the DINOv2 features and then fine-tuning the entire model.

2.1. Related Works

MeLo: Low-rank Adaptation is Better than Fine-tuning for Medical Image Diagnosis [1] aimed to explore LoRA for fine-tuning a large pretrained ViT model, specifically on medical diagnosis tasks. We base some of our implementation of LoRA on their work but adapt it for DINOv2. They added a simple linear layer on top of the ViT to classify medical images. The ViTs they used were of varying sizes (Base/Huge/Giga) and had been pre-trained on ImageNet or used CLIP to train. LoRA was leveraged by integrating LoRA layers into the attention layers much like the original LoRA paper does with transformer-based language models(LoRA 2021[2]). The tasks shown to benefit from LoRA are all medical diagnosis tasks that require classification rather than segmentation. Our work extends beyond MeLO in that we concentrate on medical segmentation tasks. Segmentation tasks are of great interest as they can be used not only to identify areas of interest from medical images but also precisely segment these areas. This precision is crucial in medical tasks such as diagnosis and tracking disease progression. Furthermore the MeLO work fine-tuned ViTs that had been trained on much less data than DINOv2, hence they do not fine-tune a proper vision foundational model. Our work's contributions are as follows:

- we show that LoRA can be used for parameter-efficient fine-tuning of a proper vision foundational model, DINOv2
- we specifically show that this LoRA based fine-tuning works for medical segmentation tasks, demonstrating its efficacy beyond classification tasks in medical diagnosis

3. EXPERIMENTS

The main objectives of our experiments are to

- Explore the potential for LoRA to improve fine-tuning time
- Explore the performance of LoRA from modality jumping from a natural image foundation model to medical image analysis

To do this we have assembled a set of tasks and compared the performance and time of fine-tuning/training

- A Linear Layer on the frozen backbone of the pre-trained model
- The whole model using a linear layer head maybe
- The model pretrained according to the LoRA procedure

Throughout this paper we will be working with DINOv2-ViT-S/14 which is a distilled version of DINOv2-ViT-g/14, for brevity we will henceforth refer to the distilled model simply as DINOv2.

3.1. Classification

We have experimented with binary classification on the [4] dataset, which consists of lung x-rays and classifies the lungs as "Normal" or "Pneumonia". We did a simple 80-20 training/testing split, then we tested the models on the test-set (234 negative samples, 390 positive samples) made by the creators of the dataset. Though the training-set was unbalanced, 1349 negative-samples 3884 positive-samples, we used a simple non-weighted crossentropyloss. Though this could have helped training

We did try to optimize the training parameters for each model, though we were not able to do as much hyper-parameter optimization as we would have liked to. We also wanted to keep loss-function and data-augmentations similar across the different experiments, since the goal is a comparative analysis and optimizing each individual model is not the main goal.

We experimented with a single linear-layer trained with the frozen DINOv2 weights, then we fine tuned the enitire model, again using a single linear layer. And at last we tried to train with LoRA for different r-values, with a linear-head.

3.2. Segmentation

In the DINOv2 paper, the authors showed that they can use frozen DINOv2 features for downstream segmentation layers using only a linear layer to classify pixels at a patch level[3]. They also demonstrate that fine-tuning the entire DINOv2 could improve task-specific performance. We use the same approach to see if we can adapt the Dino features to medical image segmentation tasks, and we also test if fine-tuning improves the performance. We train a linear layer head to classify patch level classes, and also test fine-tuning the entire model (including the linear head) with and without LoRA.

We have chosen two datasets to evaluate on. The first is the classic task of lung-segmentation in x-ray images[5]. The reason for choosing this dataset is that it serves as an easy baseline for our method. We can test initially whether our LoRA layers actually can learn new features or whether they simply confuse the model. We believe that x-ray images are close enough to natural images that DINOv2 should still be able to extract useful features, and therefore it will require little fine-tuning to achieve great performance. This makes it an easy dataset to check that LoRA actually works.

After we had achieved good results with x-ray segmentation we wanted to move to a modality that is drastically different to natural images, in order to challenge the method. We decided to test a method used for segmenting tumors (benign and malicious) in ultrasound images[6]. We discovered late in the process of working with this dataset that some of the images contained bounding-boxes or measurements around the tumor which could give the model an unfair advantage during evaluation. When visually inspecting validation predictions we see that whilst the bounding-boxes in some cases helped the segmentation in most cases they actually confused the model and led to a worse score. Still it is important to keep in mind during the results section.

We split both datasets into 80-20 training/validation, and report the validation as in DINOv2[3]. The lung segmentation has two classes one for background and one for lung. It contains 770 images. The tumor segmentation has 3 classes, background, benign, malicious. Furthermore, the tumor segmentation dataset contains a balanced amount of images with a benign tumor, images with a malicious tumor and images where there is no tumor. When evaluating either dataset we do not count background as a correct classification.

4. RESULTS

4.1. Binary classification

The models that performed best in our experiments were LoRA using a $r = 3$ and the linear layer trained on the frozen DINOv2 features. As seen in table 1 & figure 1 they performed similarly during training/evaluation, while the other two models performed similarly to each-other. The two Worst models simply predicted that all images in the test set were positives, which is likely due to the unbalanced nature of the dataset. This same behavior is observed even when using other hyper-parameters.

Timing the experiments did show a small increase in time, when using LoRA and Fine-tuning the whole model. When training the model we realized that a lot of time went into fetching/processing the images, this is due to the fact that google-colab is inefficient at retrieving data.

4.2. Segmentation

4.2.1. X-Ray Lung Segmentation

For the main experiments with the lung segmentation we use an adamW optimizer and a slowly decreasing learning rate.

Trained	Time (proportional)	Params	Acc	f1
Full-fine	1.5	22057346	0.62	0.77
LinLayer	1	770	0.72	0.90
LoRA 3	1.3	56066	0.62	0.77
LoRA 10	1.4	56066	**0.79**	**0.92**

Table 1. The final evaluation on the test-split of the pneumonia dataset. The number after LoRA represents the rank. Time is the average time per epoch. The models were trained on a google collab, using an A100 processor.

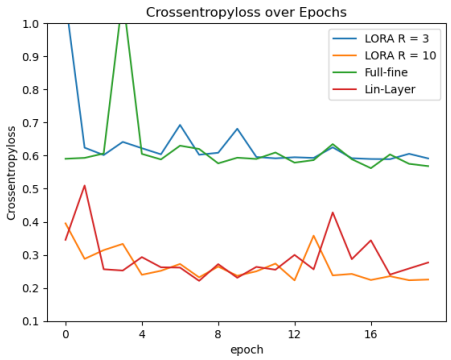

Fig. 1. Cross-entropy loss on the validation set for the different models from experiment: 1. The LoRA 3 model and the Full-fine perform similarly, while LoRA 10 and Lin-Layer, follow each other. All trainings seem to converge almost immediately indicating that the training could have been changed to obtain better results

We train for 20 epochs. In table 2 we show the results. Observe that for LoRA 10 we get a dice of 0, this would happen on occasion during training, and it seems the road to good performance is narrow. The linear layer always converged quickly to 92 percent and depending on the settings the others could converge to 92 percent. We report here the final training.

Our best model is by a narrow margin LoRA 20 which shows that we can gain performance equivalent to the linear layer with LoRA.

4.2.2. Ultrasound Tumor Segmentation

For the main experiments in the Ultrasound Tumor Segmentation we optimize using SGD and 40 epochs and a constant learning rate. During early exploration we tried using the same setup as the lung segmentation, however we discovered that our performance would always be identical to the linear layer. We attributed this first to the LoRA weights converging on zero which is why we got rid of the weight decay. This helped but we still couldn't improve the performance, so we tried using data augmentations. This gave the performance

Trained	Time(sec)	Params	Crossentropy	DICE
Full-Fine	110.8	22057346	0.39	0.63
LinLayer	91.1	770	0.06	0.92
LoRA 3	117.9	56066	0.15	0.87
LoRA 5	116.3	92930	0.16	0.86
LoRA 10	117.5	185090	0.54	0.00
LoRA 20	117.1	369410	0.06	**0.92**

Table 2. The final results of training on the lung segmentation x-ray dataset. The number after LoRa represents the rank. Time is the average time per epoch. The models were trained on a NVIDIA GeForce RTX 4070.

Trained	Time(sec)	Params	Crossentropy	DICE
Full-fine	14*	22057346	-	0.59
Lin-Layer	14.5	1155	0.17	0.52
LoRA 3	63.7	56451	0.12	**0.71**
LoRA 5	63.6	93315	0.13	0.67
LoRA 10	63.7	185475	0.14	0.63
LoRA 20	63.7	369795	0.17	0.50

Table 3. The final results of training on the ultrasound tumor segmentation dataset. The number after LoRA represents the rank. Time is the average time per epoch. The full fine-tuning was trained for 100 epochs, and used a lower learning-rate to give it a fair shot at good performance. NVIDIA GeForce RTX 4070. *the full fine-tuning was trained on an H100, so time per epoch is not an accurate comparison here.

boost we see in table 3. Notably it only improved the performance of the LoRA models, which could indicate that the extra parameters were simply data hungry. The augmentations we used were horizontal and vertical flip, as well as a random crop resize.

By using LoRA we gained an improvement of 20 points to the linear layer, and 10 points on the full fine-tuning, even though we did a larger grid search for the fine-tuning. In figure 4 we show the dice score for different ranks (denoted with an r), and in figure 3 we show two nice examples of predictions. In figure 4 we also see that the dice score is quite poor when using r=20.

5. DISCUSSION

The results from the binary-classification experiments do not prove any significant performance improvements when compared to training a linear layer, and neither do we see any significant decrease in training-time when compared to doing a full fine-tuning. The hyper-parameter optimization time-comparison was limited due to inefficiencies in the training-environment. No strong conclusions can be made on the background of these experiments.
There are a few interesting results in the segmentation part

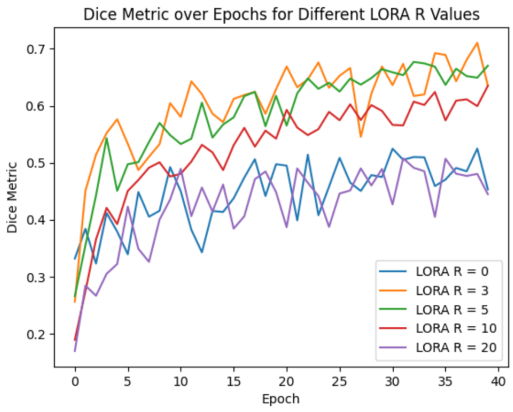

Fig. 2. Dice score for different values of r. We also include a frozen backbone to ensure that we actually gain something from training with LoRA.

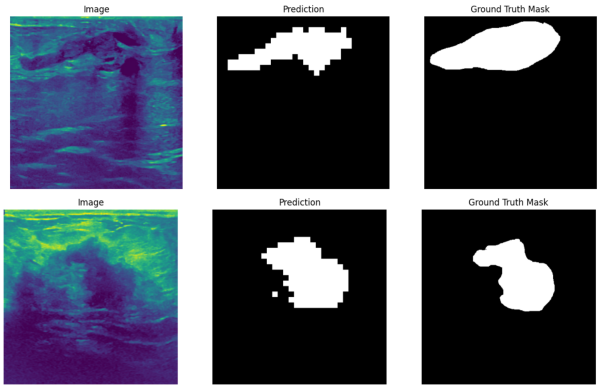

Fig. 3. Two examples of the prediction made by the r=3 LoRA model. Both examples are taken from the validation set. They illustrate that our model has become quite good at discovering even irregularly sized tumors, but still suffers from the low resolution inherent to our segmentation method.

of this assignment. Firstly we had many issues getting consistent performance when training the models for the lung segmentation. They were either on par with the linear layer or much lower, but then in a different run they would be on par with the linear layer again. The issue seems to be that the linear layer simply performs very well and the frozen features produced by DINOv2 can already be used for patch classification. This means that any perturbation of the weights will simply throw the model off and it will have to relearn useful features. This is fortunately not much of an issue as if frozen features work we can simply use them.

The more interesting dataset is the ultrasound dataset. Our results showed considerable gains when using LoRA to fine-tune, but performance decreased as we added more parameters to train. It is clear that LoRA really can help foundation models domain jump, but they seem to need a certain amount of data. We obtained good results with simple data-augmentations but it still wasn't enough for larger values of r, which is what we expected. When adding data-augmentation the full fine-tuning didn't keep up which tells us that its at least easier to fine-tune using LoRA rather then the entire model, and that LoRA requires less data. The curse of dimensionality seems to be alleviated by using LoRA for vision image transformers as well. We think this is because the LoRA fine-tuning is essentially a much simpler model but with the ability to leverage a complex network to produce the features. Essentially LoRA can alter the transformer to produce different representations but its control of the transformer is limited and hence it does not require an equal amount of data, much like smaller models require less data.

There are a few issues with the experiments we have conducted. We have not attempted to replicate our experiments with a non-foundation model as the backbone. This experiment would have told us whether we gain any benefit from fine-tuning a foundation model rather than a normally trained image model. It could be that a model trained on image net would have performed just as well with a linear layer and/or a LoRA adapter. In fact [1] does just this but for classification, but part of our project was to show that this could be achieved using a proper foundation model such as DINOv2. It does however seem unlikely that an image-net model would score 92 dice on an x-ray segmentation with just a linear layer, but we do not show explicitly that we gain a benefit from adapting DINOv2 rather than a normal image model which is a flaw of our paper. We also do not see a large improvement in training time but we do observe that we need less epochs to achieve convergence.

One of our goals were to see if there would be any improvements in training time, when using LoRA as compared to the fine-tuned model, this was not unanimously apparent from our results. This might be due to a few factors. We only trained a distilled version of the DINOv2 model, LoRA was originally created for training LLMs that are bigger than the original DINOv2-ViT-g/14, this lack of size undermines the use case of LoRA, since the dimensionality-reductions is less pronounced. Moreover for some of our experiments data fetching/processing was a major bottleneck, which further obscured any potential gains from using LoRA.

It would be interesting in the future to explore this approach for the larger DINOv2's and of course show that we gain a benefit from adapting DINO by testing a ViT that is pre-trained on ImageNet. An unintended discovery we made during these experiments, was that DINOv2's frozen features were actually quite strong even for the ultrasound images which is extremely surprising. It means that they actually

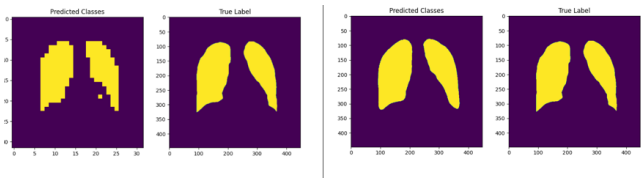

Fig. 4. Linear layer versus fine-grained adaptation.

learn features that are relevant across image modalities.

6. CONCLUSION

We have demonstrated that using LoRA when adapting a DINOv2 to the medical image domain can for some tasks benefit performance/training-time. We believe there are hints that we gain something from using LoRA on a foundation model but more experimentation is needed.

7. ADAPTATION FOR FINE-GRAINED TASKS

The linear layer trained for semantic segmentation is simple and provides good results, but it is not capable to produce high resolution segmentation maps. For achieving state-of-the-art high-resolution segmentations, the DINOv2 authors suggest using a ViT-Adapter and Mask2Former head. This however took 28 hours to train on 16 V100 GPUs. We developed and trained a much simpler UNet-like architecture to upsample the segmentations. On top of the low-res map from the trained linear layer segmentor and the image features, we learn a decoder that upsamples the resolution from 32x32 to 448x448 pixels. This improves the detail, seen by the increase in the dice metric (0.95 in the lung segmentation). The lower resolution maps guide the model, resulting in faster and better convergence.

8. REFERENCES

[1] Yitao Zhu, Zhenrong Shen, Zihao Zhao, Sheng Wang, Xin Wang, Xiangyu Zhao, Dinggang Shen, and Qian Wang, "Melo: Low-rank adaptation is better than fine-tuning for medical image diagnosis," 2024.

[2] 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," 2021.

[3] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski, "Dinov2: Learning robust visual features without supervision," 2024.

[4] Daniel S. Kermany, Kang Zhang, and Michael H. Goldbaum, "Labeled optical coherence tomography (oct) and chest x-ray images for classification," 2018.

[5] Nikhil Pandey, "Lung segmentation from chest x-ray dataset," 2019.

[6] Saba Hesaraki, "Breast ultrasound images dataset(busi)," 2023.