Assignment 1
COL828: Advanced Computer Vision
Semester I, 2024-2025.
Due Date: Sep 22, 2024

August 21

Parameter Efficient Fine-tuning of Foundation Models for Image Classification

1 Introduction

Parameter Efficient Fine-Tuning (PEFT) is a technique that allows large, pre-trained models, often referred to as foundation models, to be adapted to specific tasks with minimal modification to the original model parameters. Instead of fine-tuning the entire model, PEFT methods introduce small, task-specific layers or adjustments that require far fewer parameters to be trained. This approach is particularly valuable when dealing with large models like Vision Transformers (ViTs) or language models, where full fine-tuning would be computationally expensive and time-consuming.
In this assignment you'll implement and experiment one such technique called Visual Prompt Tuning[1]. VPT introduces only a small amount of trainable parameters in the input space while keeping the model backbone frozen. This allows to rapidly adapt the model to a new domain with reduced storage costs of updated parameters only.

2 Experiments

2.1 Data

You'll use the following data on brain-tumor classification. This dataset contains 800 images of human brain MRI images which are classified into 4 classes: *glioma*, *meningioma*, *no tumor* and *pituitary*, split as 480/320 between train and test splits. This data is a subset of the original dataset published on Kaggle[2]

2.2 Models

In this assignment, you'll experiment with CLIP ViT-base model *openai/clip-vit-base-patch16*. You're required to conduct the following experiments :

- **[E1] Zero-Shot:** Inference of base CLIP model (Note: Using an appropriate text prompt from the label is left upto your choice)
- **[E2] ViT *w*/Linear Head:** Train the Vision backbone of the CLIP model for image classification task using trainable linear head. Other parameters of the model must be frozen.
- **[E3] ViT *w*/Shallow VPT:** Utilize the technique mentioned in [1] to add a learnable token to the **input** of ViT backbone of the CLIP model keeping the remaining parameters frozen.
- **[E4] ViT *w*/Deep VPT:** Following the similar ablation in Sec3.2 in [1], modify the model in **[E3]** to add learnable tokens as input to each layer of the ViT backbone.
- **[E5] CLIP *w*/VPT:** In This experiment apply Shallow Visual Prompt Tuning to both the Vision and Text backbone of the CLIP model.
- **[E6] Full Fine-tuning ViT:** Finetune all the parameters in the Vision-Backone (ViT) of the specified CLIP model.

You are free to decide and optimize the various hyper-parameters that best suits the experiment. However the trained models must be fully converged. Use of libraries like Pytorch[3], Huggingface Transformers[4] is allowed.

2.3 Submission

Write a report with your observations upon running the above mentioned experiments, such as convergence speed, number of trainable parameters etc. You must report the training/test set accuracies along with the loss curves for both the sets for each experiment. Details Regarding submission will be shared later.

References

[1] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*, 2022.

[2] Msoud Nickparvar. Brain tumor mri dataset, 2021.

[3] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito,

Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[4] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.