Reading: Assignment Overview: Vision Transformers Using PyTorch

Estimated reading time: 2 minutes

Introduction

In this lab, you will design and implement a PyTorch-based hybrid deep learning model that integrates Convolutional Neural Networks (CNNs) with Vision Transformers (ViTs) to enhance image classification performance.

Let's go through the overview of steps and tasks that you will perform during this lab:

Steps and tasks

1. In PyTorch, to use a pre-trained model, you start by defining the CNN architecture for the pre-trained model. This deep CNN backbone transforms input images into feature maps.

```
class ConvNet(nn.Module):
    '''
    Class to define the architecture same as the imported pre-trained CNN model
    '''
    def __init__(self, num_classes):
        super().__init__()
            nn.Conv2d(3, 32, 5, padding=2), nn.ReLU(), nn.MaxPool2d(2), nn.Batch
            nn.Conv2d(32, 64, 5, padding=2), nn.ReLU(), nn.MaxPool2d(2), nn.Batc
            nn.Conv2d(64, 128, 5, padding=2), nn.ReLU(), nn.MaxPool2d(2), nn.Bat
            nn.Conv2d(128, 256, 5, padding=2), nn.ReLU(), nn.MaxPool2d(2), nn.Ba
            nn.Conv2d(256, 512, 5, padding=2), nn.ReLU(), nn.MaxPool2d(2), nn.Ba
            nn.Conv2d(512, 1024, 5, padding=2), nn.ReLU(), nn.MaxPool2d(2), nn.B
        )

    def forward_features(self, x):
        return self.features(x)      # (B,1024,H,W)
```

2. Then, you extract the "tokens" from the CNN feature map by using a patch embedding layer to convert local patches/features into tokens.
3. Next, you add a learnable classification token (cls) and positional embeddings.
4. Then, you create the ViT block with patch embedding, multi-headed self-attention, and transformer layers.
5. This ViT block is then used to create the CNN-ViT hybrid model.
6. In your first task, you will begin by creating data transforms for the training data.
7. Next, you will define the transforms for validation data. Further, your task would be to define the respective data loaders. And then, you will design and train a CNN-ViT hybrid model.
8. You will end the lab with your final task of comparing two models created and trained in the lab. The key difference between these two models is the hyperparameters used in their design. Thus, this comparison would enable you to understand the nuances of hyperparameter tuning for the best performance for your dataset.
9. After completing this lab, you will need to download and save the completed lab on your computer for final submission and evaluation at the end of this course. Good luck, and let's get started.

Note: All code displayed in the screenshots is provided within the Jupyter notebooks that you will use to complete the lab exercises.