ECE/CS/ME 539 – Fall 2024 — Activity 27

Introduction

In this session, we'll explore the differences between **Full Fine-Tuning**, **Linear Probing**, and **LoRA** (Low-Rank Adaptation) using the classic **LeNet-5** architecture as our example. We'll examine how each method affects the number of trainable parameters, computational requirements, memory usage, and discuss their respective pros and cons.

LeNet-5 Architecture

LeNet-5 is a convolutional neural network designed for handwritten and machine-printed character recognition. Below is the PyTorch implementation of LeNet-5:

PyTorch Code

```python
import torch
import torch.nn as nn

class LeNet5(nn.Module):
    def __init__(self, num_classes=10):
        super(LeNet5, self).__init__()
        self.conv1 = nn.Conv2d(1, 6, kernel_size=5)  # 1 input channel, 6 output channels
        self.relu1 = nn.ReLU()
        self.avgpool1 = nn.AvgPool2d(kernel_size=2, stride=2)
        self.conv2 = nn.Conv2d(6, 16, kernel_size=5)
        self.relu2 = nn.ReLU()
        self.avgpool2 = nn.AvgPool2d(kernel_size=2, stride=2)
        self.fc1 = nn.Linear(16*4*4, 120)
        self.relu3 = nn.ReLU()
        self.fc2 = nn.Linear(120, 84)
        self.relu4 = nn.ReLU()
        self.fc3 = nn.Linear(84, num_classes)

    def forward(self, x):
        x = self.avgpool1(self.relu1(self.conv1(x)))
        x = self.avgpool2(self.relu2(self.conv2(x)))
        x = x.view(-1, 16*4*4)
        x = self.relu3(self.fc1(x))
        x = self.relu4(self.fc2(x))
        x = self.fc3(x)
        return x
```

Questions

1. **Comparative Analysis of Fine-Tuning Methods**

   Using the LeNet-5 model provided:

   (a) **Parameter Counting:** Calculate the total number of parameters in LeNet-5 and determine how many parameters are trainable under each fine-tuning method:

   - Full Fine-Tuning
   - Linear Probing (Assuming only the last fully connected layer `fc3` is trainable)
   - LoRA (Applying LoRA only to the fully connected layers with rank $r = 2$)

   (b) **Compute and Memory Requirements:** Discuss the computational and memory requirements for training LeNet-5 under each fine-tuning method. Consider factors such as:

   - Number of operations (FLOPs) during training
   - Memory usage for model parameters and activations

2. **Pros and Cons Discussion**

   Based on your analysis from the previous question:

   (a) Discuss the pros and cons of each fine-tuning method in the context of LeNet-5.

   (b) Given a scenario where you have limited computational resources and memory but need reasonable performance, which method would you choose and why?