Ahmed Refaay

Nov. 5th, 2024

Project 1 Report: Transfer Learning with MobileNetV2 for CIFAR-10

---

1. Introduction

This project focuses on applying transfer learning techniques to the CIFAR-10 dataset using MobileNetV2, a lightweight and efficient deep learning model ideal for resource-constrained environments. The project's goals align with companies' emphasis on optimization and practical deployment in real-world scenarios.

---

2. Project Overview

- **Objective:** To develop a high-accuracy, resource-efficient model for classifying CIFAR-10 images, demonstrating skills in model customization, optimization, and practical deployment.
- **Model Choice:** MobileNetV2 was selected for its balance between accuracy and efficiency, which is essential for low-resource setups used in edge applications.

---

3. Methodology

1. **Data Loading and Preprocessing:**
    - **Dataset:** CIFAR-10, consisting of 60,000 32x32 images across 10 classes.
    - **Resizing:** Images were resized to 96x96 to be compatible with MobileNetV2.
    - **Data Augmentation:** Techniques such as random rotations, horizontal flipping, and scaling were used to enhance model generalization.
2. **Model Setup and Training:**
    - **Base Model:** Loaded MobileNetV2 without the top layers, freezing the pre-trained weights to retain learned features.
    - **Custom Layers:** Added a fully connected layer setup to tailor the model for CIFAR-10 classification.
    - **Compilation:** Used the Adam optimizer, categorical cross-entropy loss, and set metrics to track accuracy.
    - **Training and Fine-Tuning:**
        - Initially trained with frozen layers to establish a baseline.
        - Fine-tuned by unfreezing a portion of the MobileNetV2 layers to further improve performance.
    - **Outcome:** Achieved a testing accuracy of 91.89%, reflecting a robust model suitable for efficient inference.
3. **Inference and Evaluation:**

    - **Testing:** Conducted inference on new images from the internet, confirming accurate classification.
    - **Performance Metrics:** Monitored training and validation loss curves, along with accuracy trends.

---

4. Challenges and Solutions

- **Memory Management:** Encountered issues with memory exhaustion on GPU/CPU. Addressed this by using data augmentation in batches and optimizing TensorFlow settings.
- **Resource Utilization:** Explored strategies to balance CPU and GPU load, ensuring smoother training and inference processes.

---

5. Future Directions

1. **Optimization:**
    - Explore using OpenVINO or TensorRT for performance improvements on edge devices.
    - Consider model quantization or pruning to reduce the model's size and increase efficiency.
2. **Deployment:**
    - Package the model using Docker for easy deployment and integration.
    - Conduct tests in simulated production environments to ensure stability and performance.

---

6. Conclusion

This project demonstrates a thorough understanding of transfer learning and model optimization for low-resource environments. By achieving high accuracy while maintaining efficiency, the project showcases practical skills relevant to companies’ focus on high-performance computer vision solutions. The next steps include further optimization and real-world deployment practices.

Attachments:

- Full implementation code
- Training and evaluation plots (log files)
- Documentation of the deployment setup (to be prepared)