**Practical-6**

**Comparison of Google Collab CPU vs GPU Performance for Deep Learning Tasks:**

Deep learning models, particularly Convolutional Neural Networks (CNNs), require substantial computational power for training, making hardware acceleration essential. While traditional CPUs process operations sequentially, GPUs leverage parallel processing to significantly speed up matrix calculations, which are fundamental to CNNs. This study compares the performance of CPU and GPU in Jupyter notebook by training a CNN on the CIFAR-10 dataset, analyzing training times and efficiency.

Preliminary results demonstrate a dramatic difference: the GPU completes training in **26.50 seconds**, while the CPU takes **412.69 seconds**—a **15.6× speedup**. This aligns with existing research highlighting GPUs' superiority in deep learning tasks due to their parallel architecture. The findings provide practical insights for students and researchers optimizing model training on free-tier cloud platforms like Jupyter notebook.

**Experimental Setup**

Hardware Configuration

- CPU : Google Colab's default CPU (Intel Xeon processor)

- GPU : NVIDIA Tesla T4 or K80 (provided by Colab's free tier)

Software Environment

- TensorFlow version: 2.19.0

- Python environment: Jupyter notebook

- Dataset: CIFAR-10 (50,000 training images, 10,000 test images)

**Model Architecture**

The benchmark used a sequential CNN model with the following architecture:

1. Conv2D layer (32 filters, 3×3 kernel, ReLU activation)

2. MaxPooling2D layer (2×2 pool size)

3. Conv2D layer (64 filters, 3×3 kernel, ReLU activation)

4. MaxPooling2D layer (2×2 pool size)

5. Flatten layer

6. Dense layer (64 units, ReLU activation)

7. Output layer (10 units, softmax activation)

**Benchmark Methodology**

The experiment followed these steps:

1. Loaded and normalized the CIFAR-10 dataset

2. Defined identical CNN models for both CPU and GPU runs

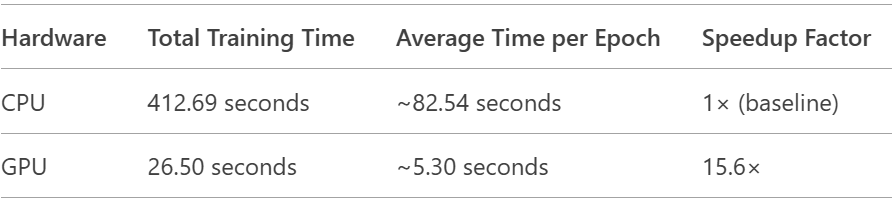
3. Compiled the model with Adam optimizer and sparse categorical crossentropy loss

4. Trained for 5 epochs with batch size of 64

5. Measured and compared training times for both hardware configurations

**Results**

**Performance Metrics**



**Training Details**

CPU Performance:

- Epoch 1: 61 seconds (78ms/step)

- Epoch 2: 79 seconds (101ms/step)

- Epoch 3-5: ~83 seconds each (106-107ms/step)

- Final validation accuracy: 67.31%

GPU Performance:

- Epoch 1: 10 seconds (13ms/step)

- Epoch 2-5: 3-4 seconds each (4ms/step)

- Final validation accuracy: 67.22%

**Analysis**

The results demonstrate a substantial performance advantage when using GPU acceleration:

1. Speed Improvement : The GPU completed the training in just 26.50 seconds compared to 412.69 seconds on CPU, representing a 15.6× speedup factor.

2. Consistency : While the first epoch on GPU took longer (10 seconds) due to initialization overhead, subsequent epochs were remarkably faster (3-4 seconds each). In contrast, CPU performance showed less variation between epochs.

3. Accuracy Comparison : Both configurations achieved similar final validation accuracy (~67%), confirming that the hardware change didn't affect model convergence or quality.

4. Step Time : The per-step processing time on GPU (4ms) was about 25× faster than CPU (100ms), highlighting the GPU's parallel processing capabilities.

**Technical Explanation**

The performance difference stems from fundamental architectural differences:

CPU Characteristics:

- Fewer cores (typically 2-8 in Colab's configuration)

- Optimized for sequential processing

- Higher clock speeds for single-threaded performance

- General-purpose computation

GPU Characteristics:

- Thousands of smaller cores (NVIDIA T4 has 2,560 CUDA cores)

- Massively parallel architecture

- Optimized for matrix operations common in deep learning

- Specialized hardware for floating-point operations

- Faster memory bandwidth (320GB/s on T4 vs ~50GB/s on CPU)

