LAB 4: STUDENT WORKSHEET

# Machine Learning Hardware Optimization

**Name:**  **Student ID:**  **Date:**

# PART 1: HARDWARE PERFORMANCE BENCHMARKING

Record the performance metrics for model training on different hardware platforms:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Hardware** | **Training Time (s)** | **Samples/Second** | **Validation Accuracy (%)** |
| CNN | CPU |  |  |  |
| CNN | GPU |  |  |  |
| FCNN | CPU |  |  |  |
| FCNN | GPU |  |  |  |
| C |  |  |  | C |

Record the performance metrics for model inference on different hardware platforms:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Hardware** | **Batch Size** | **Inference Time (ms)** | **Samples/Second** |
| CNN | CPU | 1 |  |  |
| CNN | CPU | 32 |  |  |
| CNN | GPU | 1 |  |  |
| CNN | GPU | 32 |  |  |
| FCNN | CPU | 1 |  |  |
| FCNN | CPU | 32 |  |  |
| FCNN | GPU | 1 |  |  |
| FCNN | GPU | 32 |  |  |
| C |  |  |  | C |

Based on your results:

1. Which hardware platform is most efficient for training? Why?
2. How does batch size affect inference performance? Explain the differences observed.
3. Which model architecture (CNN vs. FCNN) shows better hardware utilization? Explain.

# PART 2: MODEL QUANTIZATION

Record the performance metrics for different quantization techniques:

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy (%)** | **Model Size (MB)** | **Size Reduction (%)** |
| Original Model |  |  | N/A |
| TFLite (Float32) |  |  |  |
| TFLite (Float16) |  |  |  |
| TFLite (Int8) |  |  |  |
| C |  |  | C |

Based on your results:

1. Which quantization technique provides the best balance between model size and accuracy?
2. What is the relationship between quantization precision and model accuracy?
3. For which deployment scenarios would you recommend int8 quantization, despite potential accuracy loss?

# PART 3: MODEL PRUNING

Record the performance metrics for different pruning techniques:

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy (%)** | **Model Size (MB)** | **Size Reduction (%)** |
| Original Model |  |  | N/A |
| Pruned Model |  |  |  |
| Stripped Pruned Model |  |  |  |
| Pruned + Float16 Quantization |  |  |  |
| Pruned + Int8 Quantization |  |  |  |
| C |  |  | C |

Based on your results:

1. How effective is pruning at reducing model size while maintaining accuracy?
2. What are the combined effects of pruning and quantization?
3. What hardware benefits would you expect from a pruned model (beyond size reduction)?

# PART 4: DEPLOYMENT FORMAT COMPARISON

Record the model sizes for different deployment formats:

|  |  |  |
| --- | --- | --- |
| **Format** | **Model Size (MB)** | **Size Relative to Original Keras Model (%)** |
| Keras (H5) |  | 100% |
| TensorFlow Lite |  |  |
| ONNX |  |  |
| SavedModel |  |  |
| TensorFlow.js |  |  |
| C |  | C |

Based on your results:

1. Which deployment format is most size-efficient? Why might this be the case?
2. What are the key considerations when choosing a deployment format beyond size?
3. For a mobile deployment scenario, which format would you recommend and why?

# PART 5: COMPREHENSIVE ANALYSIS

Record the best model for different optimization priorities:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Priority** | **Best**  **Model/Technique** | **Accuracy**  **(%)** | **Size**  **(MB)** | **Accuracy Loss**  **(%)** | **Size Reduction**  **(%)** |
| Highest Accuracy |  |  |  |  |  |
| Smallest Size |  |  |  |  |  |
| Best Accuracy/Size  Trade-off |  |  |  |  |  |
| Mobile Deployment |  |  |  |  |  |
| Server Deployment |  |  |  |  |  |
| C |  |  |  |  | C |

Based on your comprehensive analysis:

1. What optimization technique provides the best accuracy-per-MB efficiency?
2. If you needed to deploy a model with <1MB size, which optimization techniques would you combine?
3. What would be your recommended approach for optimizing a real-time computer vision model for a smartphone?

# PART 6: REFLECTION

Write a short reflection (100-150 words) on what you learned about hardware optimization and its importance for ML model deployment.

# INSTRUCTOR COMMENTS

Grade: /