**1. Model Size Comparison**

|  |  |  |
| --- | --- | --- |
| **Framework** | **Model Size** | **Compression Ratio** |
| TensorFlow | 0.28 MB | 1.0x (baseline) |
| TensorFlow Lite | 88.89 KB | 0.31x smaller |
| TF Lite Quantized | 26.26 KB | 0.09x smaller |

**2. Accuracy Comparison**

|  |  |  |
| --- | --- | --- |
| **Framework** | **Test Accuracy** | **Accuracy Loss** |
| TensorFlow | 97.73% | 0.0% (baseline) |
| TensorFlow Lite | 97.73% | +0.00% |
| TF Lite Quantized | 97.71% | -0.02% |

**3. Memory Usage Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| **Framework** | **RAM Usage** | **Flash Usage** | **Total Footprint** |
| TensorFlow | 0.28 MB | 0.28 MB | 0.56 MB |
| TensorFlow Lite | 88.89 KB | 88.89 KB | 177.78 KB |
| TF Lite Micro | 40.00 KB | 26.26 KB | 66.26 KB |

## Trade-off Analysis

TensorFlow offers the highest model expressivity. It supports complex neural network structure and customization. However, it comes at the cost of higher computational demands. In conclusion, TensorFlow required much more storage space and computation power to run the model.

TFLite balances the model size and computational resources requirement by optimizing model through quantization and pruning. By reducing model size and stored parameter data type, it accelerates the model inference speed and with less deployment constraints while maintaining acceptable accuracy.

TFLite Micro is a framework tailored for microcontrollers with severe limitations. It generates the most lightweight ai model with minimal memory usage and real-time inference capabilities. By sacrificing more expressivity, models using TFLite Micro framework can achieve minimal resource consumption.

## Use Case Suitability

### **TensorFlow**: Suitable for cloud-based deployment, data centers, and scenarios requiring high model accuracy and flexibility.

### **TensorFlow Lite**: Best suited for mobile applications, IoT devices, and embedded systems where power efficiency and low latency are critical.

### **TensorFlow Lite Micro**: Designed for microcontrollers and extremely resource-constrained environments.

## Quantization Impact

* **Model Accuracy:** Slightly reduce model accuracy caused by precision loss.
* **Model Size:** Significantly reduces model size
* **Inference Speed:** Enhances inference speed by enabling faster computations and hardware acceleration.
* **Development Complexity:** Adds complexity to the development process, requiring additional steps in quantization-aware training

## Deployment Considerations

* **Hardware Compatibility:** Availability of hardware accelerators and support for specific hardware devices
* **Update:** Some frameworks will have more frequent updates on the newest technology and bug fixes. Using those frameworks will ensure a better develop experience.

## Future Directions

* **Hardware-Aware Optimization:** Frameworks will increasingly incorporate hardware-specific optimizations, enabling models to leverage unique capabilities of emerging accelerators and microarchitectures.
* **Edge-Cloud Integration:** Maybe someday a framework can communicate seamlessly and run a larger model together.
* **Support for New Hardware Paradigms:** As quantum computing keeps developing, frameworks will adapt to support these novel architectures.

### Resolver notes:

 Identified the required operators: From the interpreter feedback

Why you chose this approach: ALL\_OP\_RESOLVER returns error when building