TinyML Ops: Overview, Challenges and Implementation

Avyay Sah¹,², Soham Chatterjee², Archana Vaidheeswaran²
Purdue University¹,²
ScaleDown Team, Singapore²

TinyML Ops are a set of best practices that can help you build and deploy machine learning applications on TinyML devices successfully.

## TinyML Ops Architecture

**On Host**
- **Train**: Model training
- **Optimize**: Model optimization
- **Model**: Model deployment
- **Benchmark**: Model benchmarking

**On Device**
- **Security and Robustness**: Security and robustness measures
- **Monitor and Update**: Model monitoring and updates
- **Deploy**: Model deployment

## Model Optimization

- **Tools and Techniques**
  - Optimization
  - Proxy Metrics

## Challenges

- Before deploying models, they need to be optimized to get the best performance from the target hardware.
  - Optimization may not reduce model size and operations.
  - Models may also be optimized for energy consumption, peak RAM usage, latency, throughput etc.
  - Optimization can reduce accuracy and increase latency.

## Model Optimization Techniques

- **Quant**: Quantization
- **Pruning**: Pruning
- **Weight Sharing and Learning**: Weight sharing and learning

## Tools

- **TFLite**: TensorFlow Lite
- **OpenVINO**: OpenVINO
- **ONNX**: ONNX

## Technologies

- **Architecture and Operations**: Architecture and operations for TinyML devices
- **Scalability**: Scalability of TinyML models
- **Security**: Security aspects of TinyML

## Deployment Architectures

- **Simple Operations and Layers**: Separable Convolutions
- **Networks**: MobileNet and EfficientNet

## Scaling

- **Difficult after deployment**: Scaling is difficult after deployment.

## Proxy Metrics

- **Robustness**: Robustness is an important proxy metric.

## Knowledge Distillation

- **Modeling**: Modeling of perceptor metrics.

## TinyML Devices

- **Partial updates**: Partial updates should be possible to reduce battery consumption.

## TinyML Devices Need Updates for

- **Software and Hardware**: Updates are difficult to do.

## Challenges

- **Deployment Architectures**
  - Models are deployed on resource-constrained edge devices.
  - Small models with few supported architectures and Ops.
  - Latency, throughput, power consumption are important.
  - No containers; logging and monitoring is difficult to do.
  - Performance checks and model updates are difficult to do.
  - Scaling is difficult after deployment; Robustness is important.

## TinyML Ops: Overview, Challenges and Implementation

**TinyML Ops** refers to best practices that can help you build and deploy machine learning applications on TinyML devices successfully.

- **Models are trained and deployed on powerful devices.**
- **Large models with multiple supported architectures and Ops.**
- **Accuracy and availability are important.**
- **Containerization, CI/CD, logging and monitoring is required.**
- **Performance checks and model updates are done regularly.**
- **Robustness and Autoscaling to traffic.**

## Challenges

- **Supported architectures and Ops.**
- **Models are deployed on resource-constrained edge devices.**
- **Latency, throughput, power consumption are important.**
- **No containers; logging and monitoring is difficult to do.**
- **Performance checks and model updates are difficult to do.**
- **Scaling is difficult after deployment; Robustness is important.**

## Deployment Architectures

- **Simple Operations and Layers**: Separable Convolutions
- **Networks**: MobileNet and EfficientNet

## Scaling

- **Difficult after deployment**: Scaling is difficult after deployment.

## Proxy Metrics

- **Robustness**: Robustness is an important proxy metric.

## Knowledge Distillation

- **Modeling**: Modeling of perceptor metrics.

## TinyML Devices

- **Partial updates**: Partial updates should be possible to reduce battery consumption.

## TinyML Devices Need Updates for

- **Software and Hardware**: Updates are difficult to do.

## Challenges

- **Deployment Architectures**
  - Models are deployed on resource-constrained edge devices.
  - Small models with few supported architectures and Ops.
  - Latency, throughput, power consumption are important.
  - No containers; logging and monitoring is difficult to do.
  - Performance checks and model updates are difficult to do.
  - Scaling is difficult after deployment; Robustness is important.

## TinyML Ops: Overview, Challenges and Implementation

- **Models are trained and deployed on powerful devices.**
- **Large models with multiple supported architectures and Ops.**
- **Accuracy and availability are important.**
- **Containerization, CI/CD, logging and monitoring is required.**
- **Performance checks and model updates are done regularly.**
- **Robustness and Autoscaling to traffic.**

## Challenges

- **Supported architectures and Ops.**
- **Models are deployed on resource-constrained edge devices.**
- **Latency, throughput, power consumption are important.**
- **No containers; logging and monitoring is difficult to do.**
- **Performance checks and model updates are difficult to do.**
- **Scaling is difficult after deployment; Robustness is important.**

## Deployment Architectures

- **Simple Operations and Layers**: Separable Convolutions
- **Networks**: MobileNet and EfficientNet

## Scaling

- **Difficult after deployment**: Scaling is difficult after deployment.

## Proxy Metrics

- **Robustness**: Robustness is an important proxy metric.

## Knowledge Distillation

- **Modeling**: Modeling of perceptor metrics.

## TinyML Devices

- **Partial updates**: Partial updates should be possible to reduce battery consumption.

## TinyML Devices Need Updates for

- **Software and Hardware**: Updates are difficult to do.

## Challenges

- **Deployment Architectures**
  - Models are deployed on resource-constrained edge devices.
  - Small models with few supported architectures and Ops.
  - Latency, throughput, power consumption are important.
  - No containers; logging and monitoring is difficult to do.
  - Performance checks and model updates are difficult to do.
  - Scaling is difficult after deployment; Robustness is important.

## TinyML Ops: Overview, Challenges and Implementation

- **Models are trained and deployed on powerful devices.**
- **Large models with multiple supported architectures and Ops.**
- **Accuracy and availability are important.**
- **Containerization, CI/CD, logging and monitoring is required.**
- **Performance checks and model updates are done regularly.**
- **Robustness and Autoscaling to traffic.**

## Challenges

- **Supported architectures and Ops.**
- **Models are deployed on resource-constrained edge devices.**
- **Latency, throughput, power consumption are important.**
- **No containers; logging and monitoring is difficult to do.**
- **Performance checks and model updates are difficult to do.**
- **Scaling is difficult after deployment; Robustness is important.**