MERCon 2025
MORATUWA ENGINEERING RESEARCH CONFERENCE

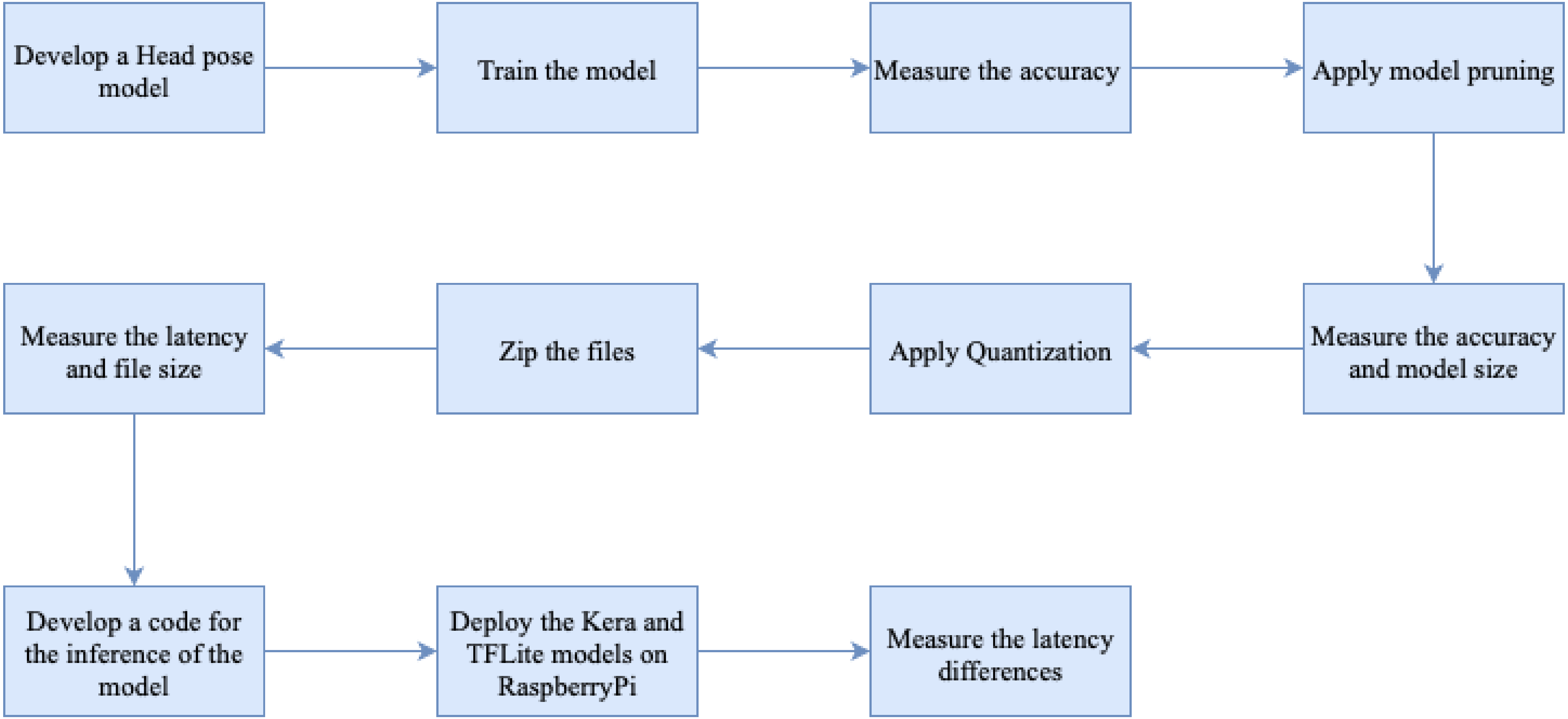
TinyML: A Compact Revolution in Engineering AI

Session 3.1 : Hands-on session on Model Pruning and Quantization for head pose estimation using Tensorflow

Asiri Gawesha Lindamulage
Open University of Sri Lanka

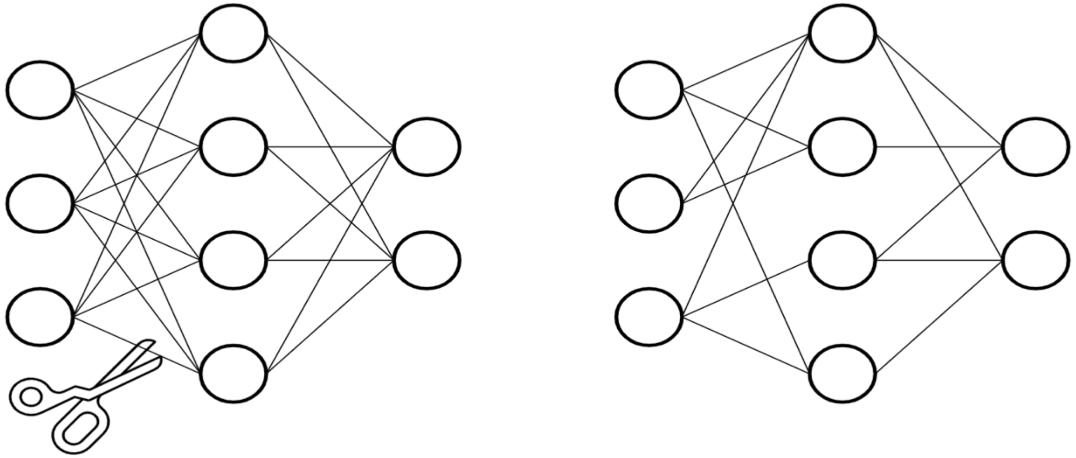
THE OPEN UNIVERSITY OF SRI LANKA

End-to-end model deployment pipeline



What is model pruning?

- Model pruning removes unnessary weights from the neural network to,
  - Improve the inference time
  - Reduce File size
- Weights almost zero below a given threshold ->0
- Model purning has two types,
  - Dynamic pruning
  - Static pruning



Before pruning After pruning

$$s_t = s_f$$

where

s_t = Current sparsity

s_f = final sparsity

Static sparsity equation

$$s_t = s_f + (s_i - s_f)\left(1 - \frac{t - t_0}{(t_f - t_0)\Delta t}\right)^3,$$

where

s_t = Current sparsity value s_i = Initial sparsity

s_f = final sparsity t_0 = Starting training step

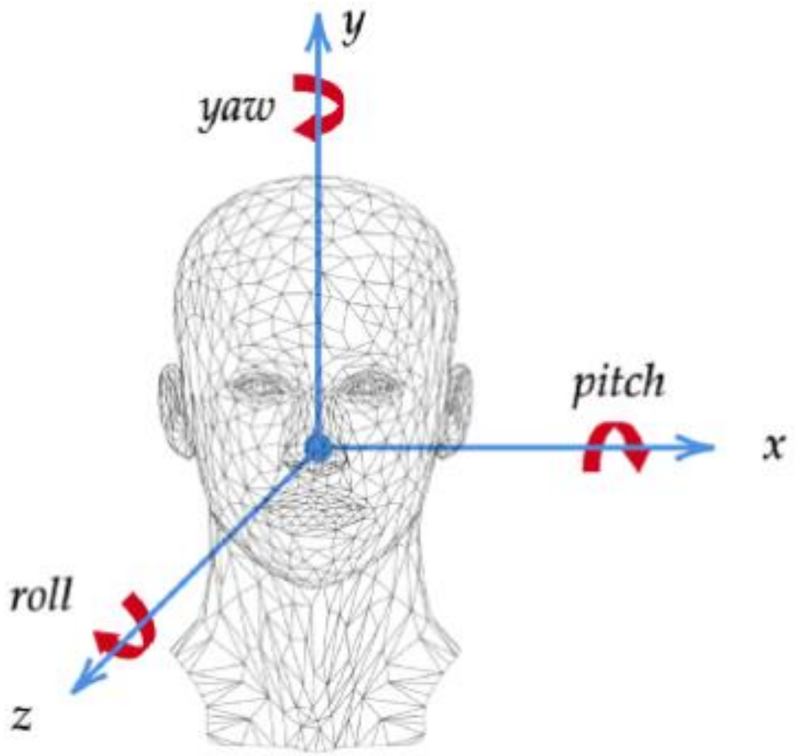
t_f = Ending training step Δt = Pruning frequency

Dynamic sparsity equation

Why Head Pose Estimation

Used in many applications

- AR/VR headsets and wearables
- portable assistive technologies
- Driver Monitoring Systems



Down Stream Tasks

- Gaze Estimation
- Human-Computer Interaction
- Action Recognition

- Regression method
- The Mean Absolute Error is calculated by averaging the errors of the three angles.

$$\text{MAE} = \frac{1}{3}\left(|y - \hat{y}| + |p - \hat{p}| + |r - \hat{r}|\right)$$

- Benchmarking Datasets- BIWI, AFLW2000, 300WLP

Aspect	Arduino Nano 33 BLE Sense	Raspberry Pi 4/5	Normal PC (Laptop/Desktop)
Processor	ARM Cortex-M4 @ 64 MHz	ARM Cortex-A72 (quad-core @ ~1.5,Äì2 GHz)	Intel/AMD x86 (multi-core, ~3+ GHz)
RAM	256 KB SRAM	1,Äì8 GB LPDDR4	4,Äì64 GB DDR4/DDR5
Flash/Storage	1 MB Flash	8,Äì128 GB microSD or SSD	256 GB ,Äì 2 TB SSD/HDD
ML Capability	TinyML models only	Quantized and small full ML models	Full-scale ML/DL (CNNs, Transformers
Model Format	TFLite Micro (.tflite)	TFLite / ONNX / PyTorch (lightweight)	TensorFlow, PyTorch, ONNX, JAX
Real-Time Support	Excellent (<1ms latency)	Moderate (10,Äì100ms latency)	Variable (not real-time by design)
Power Consumption	~20 mW	~2.5,Äì10 W	~30,Äì300+ W
Battery Operable	Yes (coin cell or LiPo)	Yes (power bank)	Limited (laptop), No (desktop)

Some experimental results

- Model Used- EfficientNet(B0) based Headpose Estimation model [1]
- Trained on BIWI Kinect Head Pose [13], 300W-LP [14] and tested on AFLW 2000 [14]
- 8.63 Mean absolute error (MAE) with a file size of 110MB
- Analyse the parameter selection for model pruning
  - Static pruning
    - Static sparsity - 50%, 75% and 87.5%
    - Starting step (epoch) - 0, 20, 40, 60, 80
    - Ending step (epoch) – 20, 40, 60, 80
  - Dynamic pruning
    - Final sparsity – 50%, 75%, and 87.5%
    - Starting step (epoch) - 0, 20, 40, 60, 80
    - Ending step (epoch) – 20, 40, 60, 80
- Combine pruning with post optimizers
  - Experimental post pruning optimizer
  - Dynamic post quantization

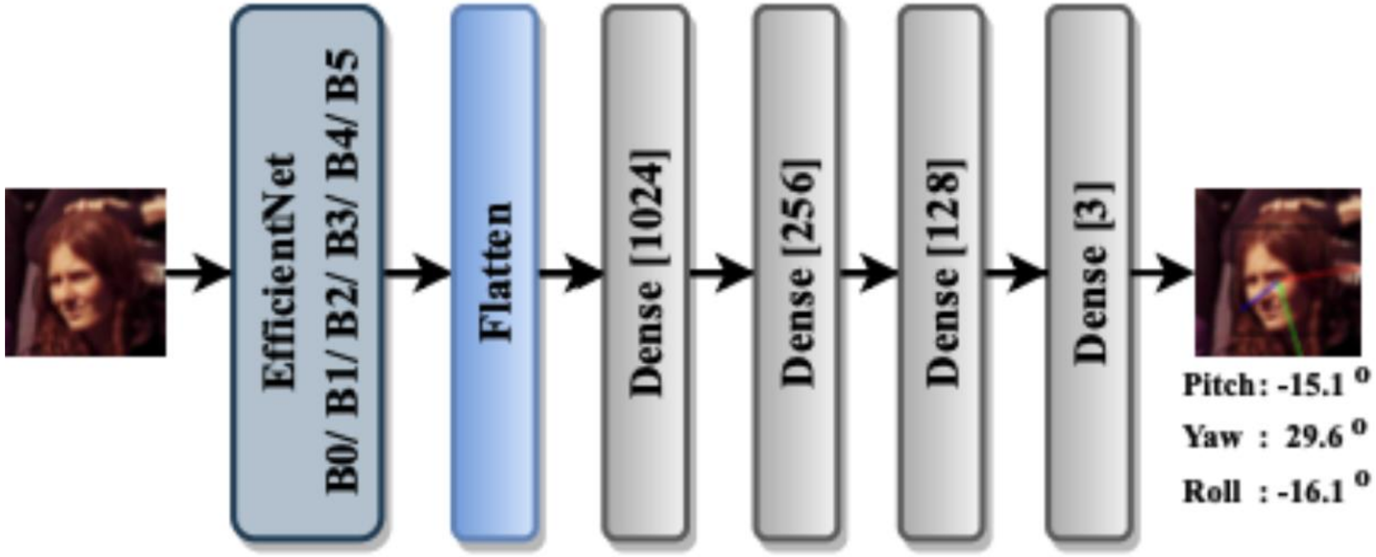


Figure 1. Model Architecture (source-[12])

MAE and model size variation with sparsity percentage

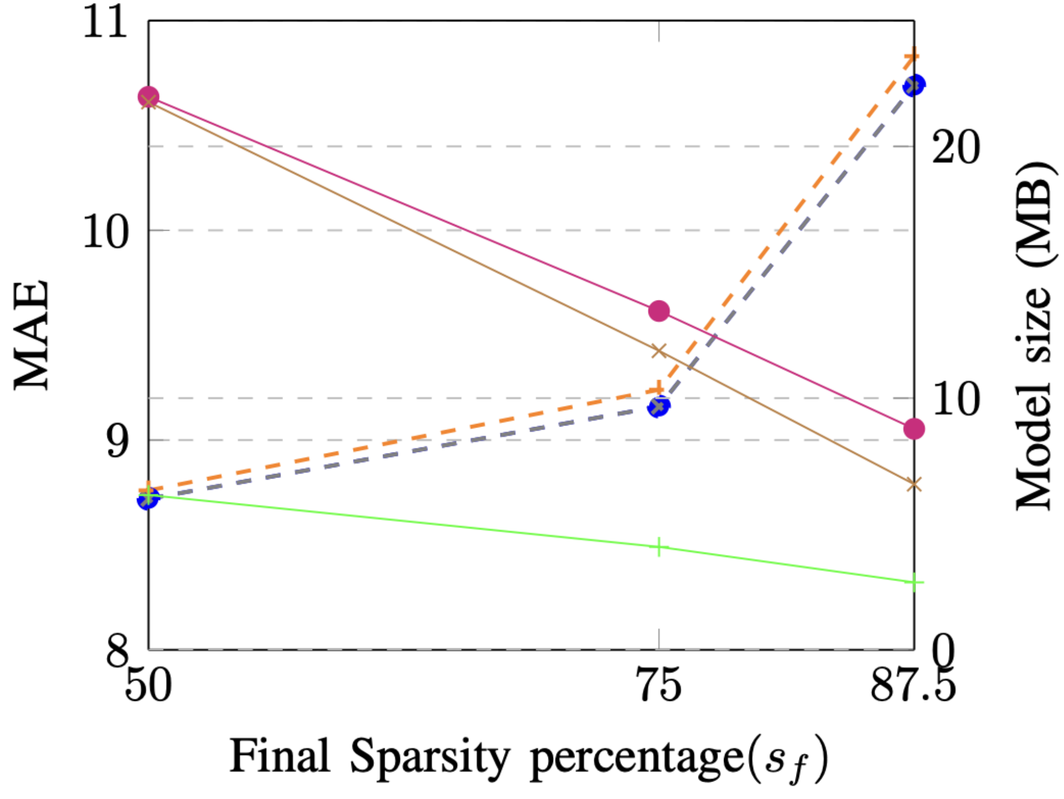
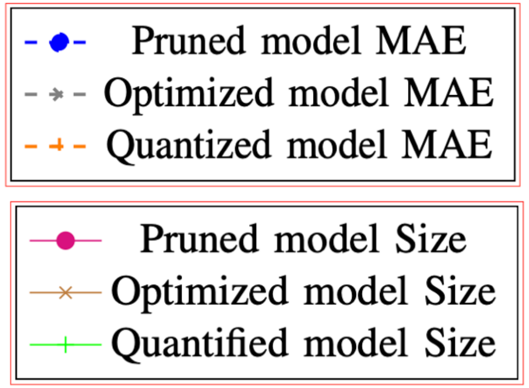


Figure 1. Dynamic Pruning



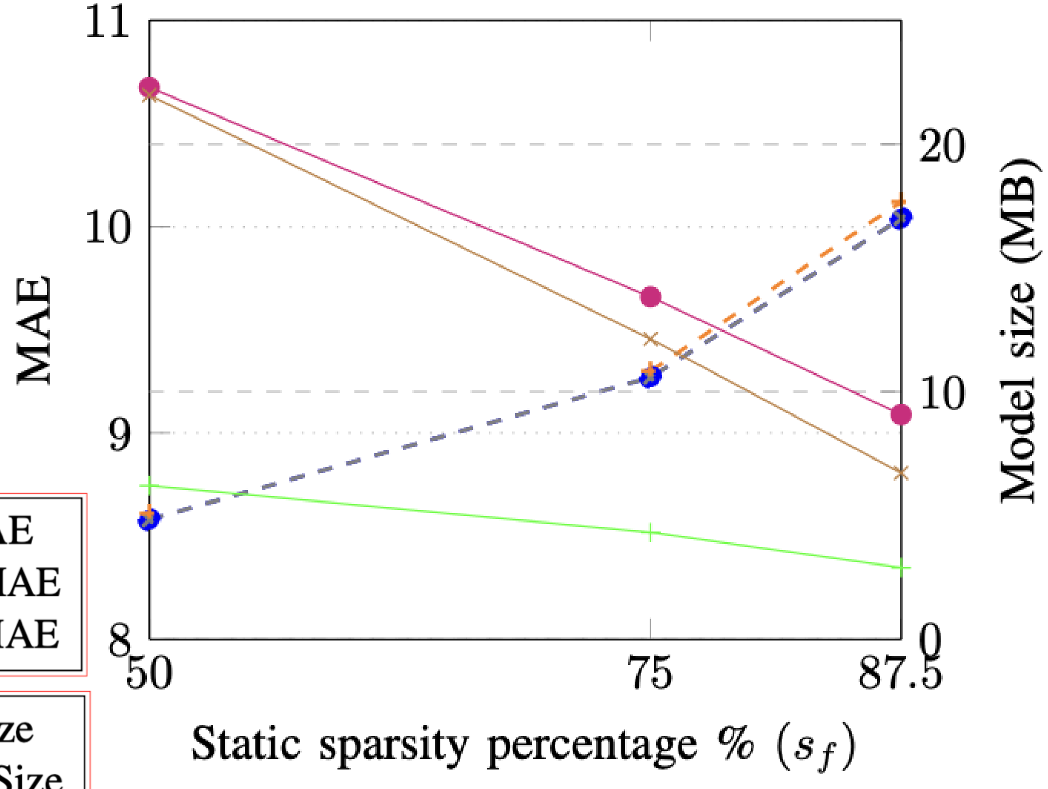
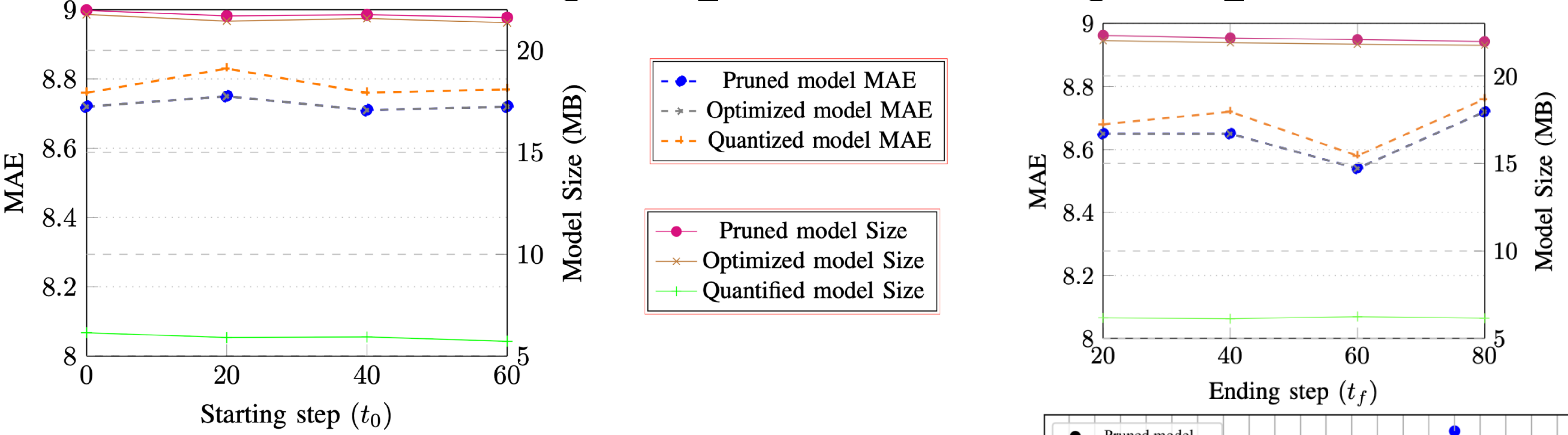


Figure 2 .Static Pruning

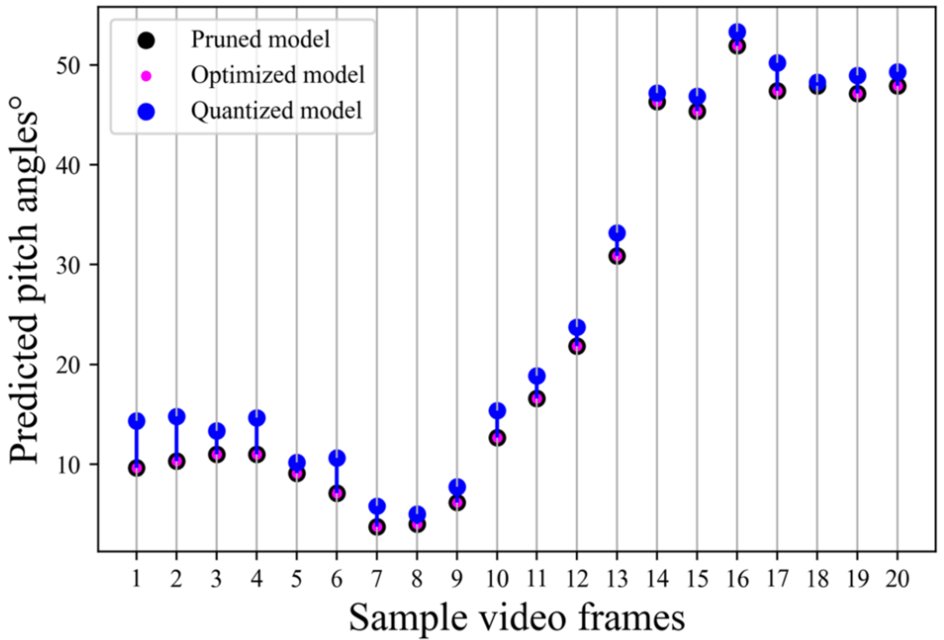
- Sparsity percentage↑ -> MAE ↑ - Accuracy ↓ - File size↓
- Post purning optimizer reduces file size without altering the model accuracy
- Dynamic quantization can reduce file size significantly with a slightly reduced model accuracy

Starting step and ending step



- Best starting point = 0
- Best ending point = 60 (3/4 of the total training steps)
- Static and dynamic pruning behaviour is identical

- Post optimizers:
- Post pruning optimizer reduces model size around 3 MB without altering model accuracy
- Dynamic post quantizer reduces the file size upto 15MB while sacrifising model accuracy below 0.1 (MAE)



Summary of the results

BEST MODELS OF DYNAMIC AND CONSTANT PRUNING

Model	Training parameters				Pruned model		Post pruned model		Post quantized model	
	Initial Sparsity	Final Sparsity	Starting step	End step	MAE	Size (MB)	MAE	Size (MB)	MAE	Size (MB)
Dynamic(Best Accuracy)	0.00	0.50	0.00	60.00	8.54	22.08	8.54	21.82	8.58	6.24
Dynamic(Best model size)	0.00	0.88	40.00	80.00	12.31	9.22	11.98	8.58	12.17	2.54
Constant((Best Accuracy)	0.00	0.50	60.00	80.00	8.57	35.23	8.57	35.24	8.59	7.85
Constant(Best model size)	0.00	0.88	60.00	80.00	11	9.21	12.39	8.41	12.51	2.59

- Starting step and ending step – best combination 0 to 60.
- Final sparsity – final sparsity higher -> lower the model accuracy
  - If priority -> model size - 87.5%
  - If priority-> model accuracy - 50%
- File size can be further reduced by combining with dynamic post- quantization sacrificing model accuracy

Source: Comparative Study of Parameter Selection for Enhanced Edge Inference for a Multi-Output Regression model for Head Pose Estimation (Tencon 2022)

TinyML: A Compact Revolution in Engineering AI

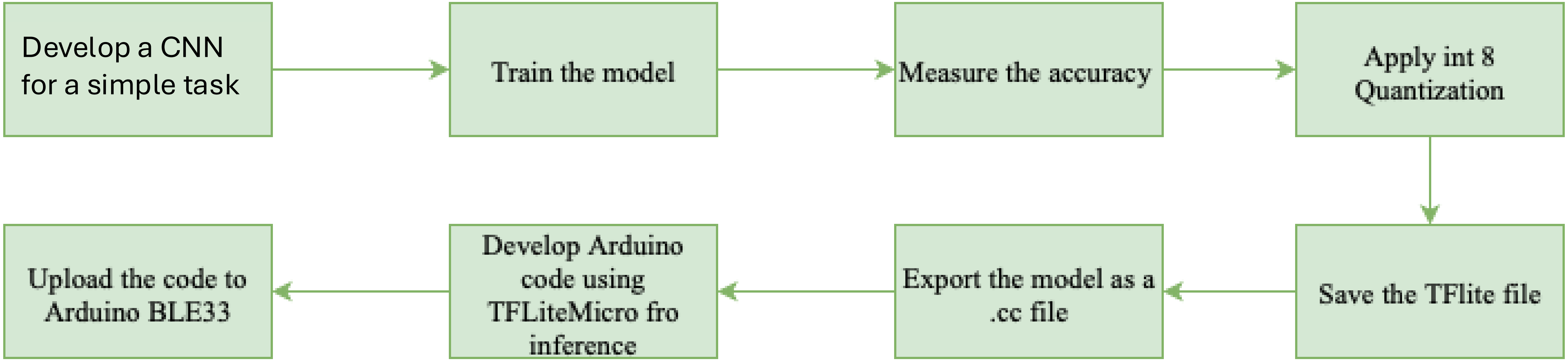
Session 3.2 : Hands-on session on deploying a simple CNN on Arduino BLE 33

Asiri Gawesha Lindamulage

Open University of Sri Lanka

aglin@ou.ac.lk

Arduino BLE 33 Deployment Cycle



Some very recent structured model pruning techniques

Some of the key turning points in developement of model pruning techniques

Theories	Authors
To prune, or not to prune: exploring the efficacy of pruning for model compression – Unstructured pruning method.	Michael Zhu, Suyog Gupta
THE LOTTERY TICKET HYPOTHESIS (2019) – finds a smaller architecture hidden within the original model	Jonathan Frankle, Michael Carbin
DepGraph: Towards Any Structural Pruning (2023) - based on the connections between layers and its been applied to GNNs and CNNS	Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, Xinchao Wang
Optimized Transformer Models: l′BERT with CNN-like Pruning and Quantization(2024)	Muhammad Hamis Haider, Stephany Valarezo-Plaza, Sayed Muhsin
ARPruning: An automatic channel pruning based on attention map ranking(2024)	Tongtong Yuan , Zulin Li , Bo Liu , Yinan Tang , Yujia Liu

Experimental possibilities

- Combine model pruning with quantization.
- Try structured model pruning methods available on PyTorch
- Develop structured model pruning techniques