# Ternary Hybrid Neural-Tree Networks for Highly

# Constrained IoT Applications

Dibakar Gope, Ganesh Dasika, Matthew Mattina **Arm ML Research Lab** 

## Challenge

 ML algorithms are increasing deployed at the edge in IoT devices



Intelligent

Assistants









- These devices are highly constrained in both memory and compute budget
- Aggressive model compression is required to
- -Target severely constrained microcontrollers
- -Deploy more IoT applications on them



**BBC Micro:Bit** Arm Cortex M0 (16KB SRAM)



LPCXpresso 1125 Arm Cortex M0 (8kB SRAM)

#### **Prior Solutions**

- Architectural optimization
- Depthwise separable (DS) convolution
- Bonsai decision trees [1]
- Model quantization
- Binary/ternary quantization
- StrassenNets [2]
- Model pruning
- Low-rank matrix factorization

#### Problem

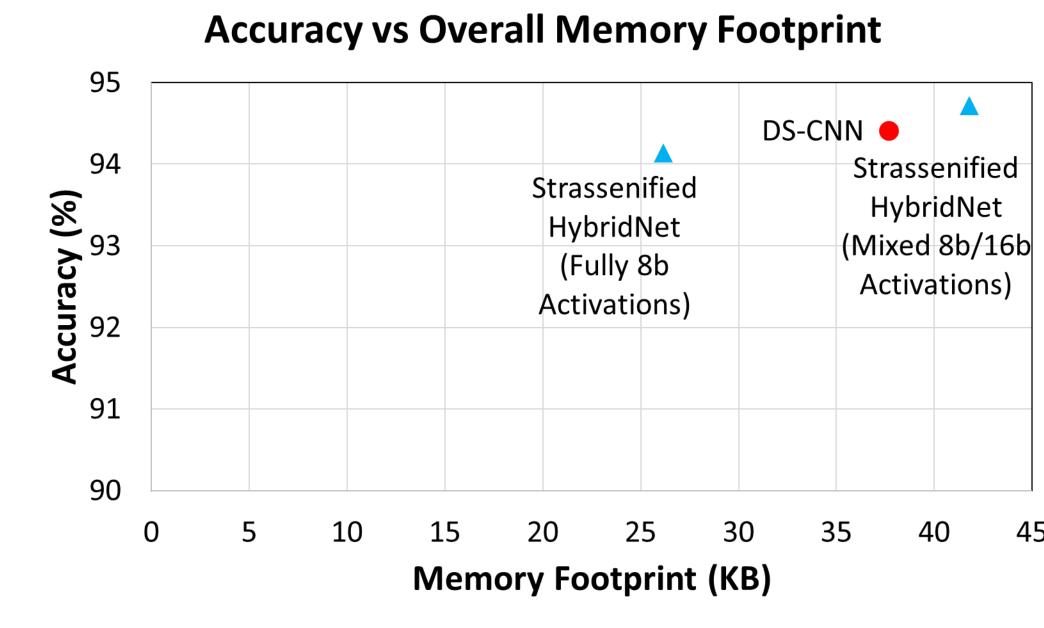
- cannot offer compression than the state-of-the-art for popular IoT applications like keyword spotting (KWS)
- They come with their own advantages and limitations

#### **Observations with Prior Solutions Accuracy vs Model Size** MFCC Features Conv1 DS-Conv1 Hidden Convolutiona DS-Conv2 Layer **Accuracy vs #Operations** DS-Conv3 DS-Conv4 r=0.5Cout Strassen Convolution Pooling layer 98% reduction in MULs Strassenified DS-CNN Modest savings in model size Output layer **Operations (M)** >50% increase in ADDs/Ops to achieve iso accuracy State-of-the-art DS-CNN. r = width of hidden layer **Accuracy vs Model Size** for KWS [3] MFCC Features Shape: TxF P=128,D=4 P=64,D=2 P=128,D=2 Bonsai tree observes poor prediction Bonsai Tree accuracy for KWS Bonsai tree → unable to extract rich **Model Size (KB)** features from KWS **Accuracy vs #Operations** Large model size $\begin{pmatrix} \mathbf{W}_2, \mathbf{V}_2 \\ \mathbf{\theta}_2^t \hat{\mathbf{x}} > \end{pmatrix}$ Low computational costs than neural P = feature width, D = depth of Bonsai tree DS-CNN P=128,D=4 P=64,D=4 P=128,D=2 $(\mathbf{W}_5, \mathbf{V}_5)$ $(\mathbf{W}_6, \mathbf{V}_6)$ Bonsai Tree **Bonsai Decision Tree** Operations (M) Our Solution: Hybrid Neural-Tree Network MFCC Features Shape: TxF **DS-CNN vs.** Strassenified **DS-CNN** Conv1 vs. Bonsai Tree DS-Conv1 Strassenified DS-CNN DS-Conv2 DS-CNN Bonsai Tree Model Size (KB) $\mathbf{W}_{1}^{\mathrm{T}}\widehat{\boldsymbol{D}}$ tanh $(\mathbf{V}_{1}^{\mathrm{T}}\widehat{\boldsymbol{D}})\setminus$ $+\mathbf{W}_{2}^{\mathrm{T}}\widehat{\boldsymbol{D}} \tanh(\mathbf{V}_{2}^{\mathrm{T}}\widehat{\boldsymbol{D}})$ $+\mathbf{W}_{4}^{\mathrm{T}}\widehat{\mathbf{D}}$ tanh $(\mathbf{V}_{4}^{\mathrm{T}}\widehat{\mathbf{D}})$

### **Evaluation Results**

- 11.1% reduction in Ops, 98.9% reduction in MULs, 12.2% reduction in ADDs
- 52.2% reduction in model size, 30.6% reduction overall memory footprint (memory footprint = model size + size of intermediate activations)
- 0.27% loss in model accuracy
- Activations quantized to 8b fixed-point format
- 0.27% loss is attributed to high sensitivity of strassenified depthwise layers towards 8b quantizations

## **Accuracy vs #Operations** DS-CNN Strassenified Strassenified HybridNet DS-CNN (r=0.75Cout) **Operations (M)**



Accuracy, operations, and model size of hybrid network and improvement over state-of-the-art KWS networks

### References

- [1] Kumar et al., "Resource-efficient Machine Learning in 2 KB RAM for the Internet of Things", ICML 2017
- [2] Tschannen et al., "StrassenNets: Deep Learning with a Multiplication Budget", ICML 2018
- [3] Zhang et al., "Hello Edge: Keyword Spotting on Microcontrollers", 2017