Energy-Efficient Speaker Identification Using Low-Precision Networks

ICASSP 2018

Skanda Koppula, Jim Glass, A.P. Chandrakasan

Massachusetts Institute of Technology

Outline

1. Motivation and Prior Work

2a. Smorgasboard of Techniques a Small-Footprint Speaker ID
   a. Normalization Folding
   b. [-1,1]-Fixed Point Quant.
   c. Trained Ternary Quant.
   d. Model Pruning and Distillation

2b. Experiments on RSR2015

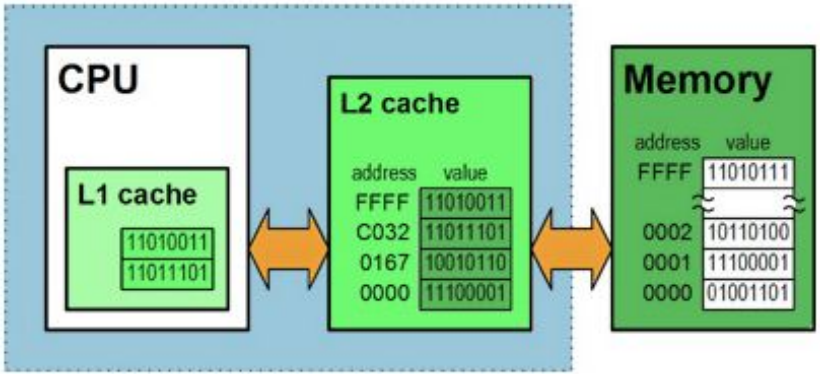
3. Translating Model Evaluation to FPGA

4. Conclusion

Motivating Small Footprint Speaker ID

- Recent work to match human benchmarks have made models wider, deeper, and increased connectivity
  - Large networks (e.g. VGG-X/Listen-Attend-Spell/Tacotron) are upwards of 100+ MB for parameter storage
- Embedded devices (wearables, sensors) have memory and energy constraints (flash on some microcontrollers < 1MB)
- Speaker identification useful for authenticating users and is performed on-device for privacy and security reasons

Tiny Models Have Energy and Latency Benefits!



i5 L2 Cache: ~1 MB
i7 L2 Cache: ~1.5 MB

Operation	Energy [pJ]	Relative Cost
32 bit int ADD	0.1	1
32 bit float ADD	0.9	9
32 bit int MULT	3.1	31
32 bit float MULT	3.7	37
32 bit 32KB SRAM	5	50
32 bit DRAM	**640**	**6400**

Application Specific Integrated Circuit
(65nm CMOS, DDR3 DRAM)

Table from Han et al., ICLR 2016

Prior Work Has Focused on Model Architecture Search

- End-to-end speaker ID wary of parameter explosion:
  - FCN[1,2]/CNN[3]/LCN[3]/Depthwise-Separable Conv[4]/Maxout[5] with Batch Norm ~ 10MB
  - log-mel features to speaker classification softmax or d-vector
- Hardware-focused SID:
  - On FPGA: systems that use SVM[6] and GMM-UBM[7]
- Recent work on quantization and compression:
  - Huffman-encoding parameters[8], pruning[8], quantization[9,10]

[1] Snyder et al. IEEE SLT 2016 [2] Kenny et al. Odyssey 2016 [3] Parada et al. Interspeech 2015 [4] Howard et al. ICML 2017 [5] Variani ICASSP 2014 [6] Saha PCS 2010 [7] Kan et al. 2010 Reconfigurable Computing [8] Han et al., ICLR 2016 [9] Courbarioux et al. NIPS 2016 [10] Zhou et al. ICLR 2016

Prior Work Has Focused on Model Architecture Search

- End-to-end speaker ID wary of parameter explosion:
  - FCN[1,2]/CNN[3]/LCN[3]/Depthwise-Separable Conv[4]/Maxout[5] with Batch Norm ~ 10MB
  - log-mel features to speaker classification softmax or d-vector
- Hardware-focused SID:
  - On FPGA: systems that use SVM[6] and GMM-UBM[7]
- **Recent work on quantization and compression:**
  - **Huffman-encoding parameters[8], pruning[8], quantization[9,10]**

[1] Snyder et al. IEEE SLT 2016 [2] Kenny et al. Odyssey 2016 [3] Parada et al. Interspeech 2015 [4] Howard et al. ICML 2017 [5] Variani ICASSP 2014 [6] Saha PCS 2010 [7] Kan et al. 2010 Reconfigurable Computing [8] Han et al., ICLR 2016 [9] Courbarioux et al. NIPS 2016 [10] Zhou et al. ICLR 2016

Our four steps for tiny models:

- Normalization Folding
- Quantization
  - [0,1]-FxPt Quantization
  - Trained Ternary Quantization
- Model Pruning
- Digital Accelerator Architecture

Measuring models with a constructed metric:

$$\texttt{score} = \log_{10}(\#\text{multiplies} \times \text{bytesize} \times \text{classification error})$$

Initial Speaker ID Setup

- Closed Speaker Set, Text-Dependent (softmax error, not d-vector)
- Size-20 Log-Mel Feature Input, 25 Frame Context
- Averaging classifications across the utterance

- RSR2015 (255 speakers, 100 unique text utterances)
- Prior Work: FCN, CNN, Maxout, Depth-wise Seperable Conv.

FCN/CNN/LCN Baselines Have Strongest Performance

Model	Error	Mults	Parameters	Score
Fully Connected, Small	0.093959	519K	520K	10.404
Fully Connected, Large	0.029713	1399K	1399K	10.763
Convolutional	0.12574	266K	260K	9.936
Locally Connected	0.11322	276K	287K	9.954
Maxout, Small	0.36653	1821K	1822K	12.084
Maxout, Large	0.26322	2134K	2133K	12.077
Depth-Seperable Conv., Small	0.11144	1245K	1227K	11.225
Depth-Seperable Conv., Large	0.29555	368K	335K	10.548

“Normalization Folding” Trims BN Parameters

- Batch normalization between layers of network:

$$x_{norm} = \gamma \frac{x - \mu}{\sqrt{\sigma}} + \beta$$

$$y = f(W x_{norm} + b)$$

- We “fold” norm into the second equation:

$$W' = \gamma \frac{W}{\sqrt{\sigma}} \qquad b' = b + W\left(\beta - \frac{\gamma \mu}{\sqrt{\sigma}}\right)$$

$$y = f(W'x + b')$$

Norm. Folding Offers Incremental Improvements

Model	Error	Mults	Params	Score	Baseline Score
Fully Connected, Small	0.093962	517K	519K	10.402	10.404
Fully Connected, Large	0.029721	1394K	1396K	10.762	10.763
Convolutional	0.12604	263K	258K	9.935	9.936
Locally Connected	0.11410	274K	286K	9.956	9.956

~1% decrease in model size, multiplications, and score □

[-1,1]-Fixed Pt. Quant. For Parameter Downsizing

- Apply quantizing op within the net's training graph:

$$x_{[0,1]} = \frac{x}{2 \max |x|} + \frac{1}{2}$$

$$x_{quant} = 2 \ \mathtt{quant}(x_{[0,1]}) - 1$$

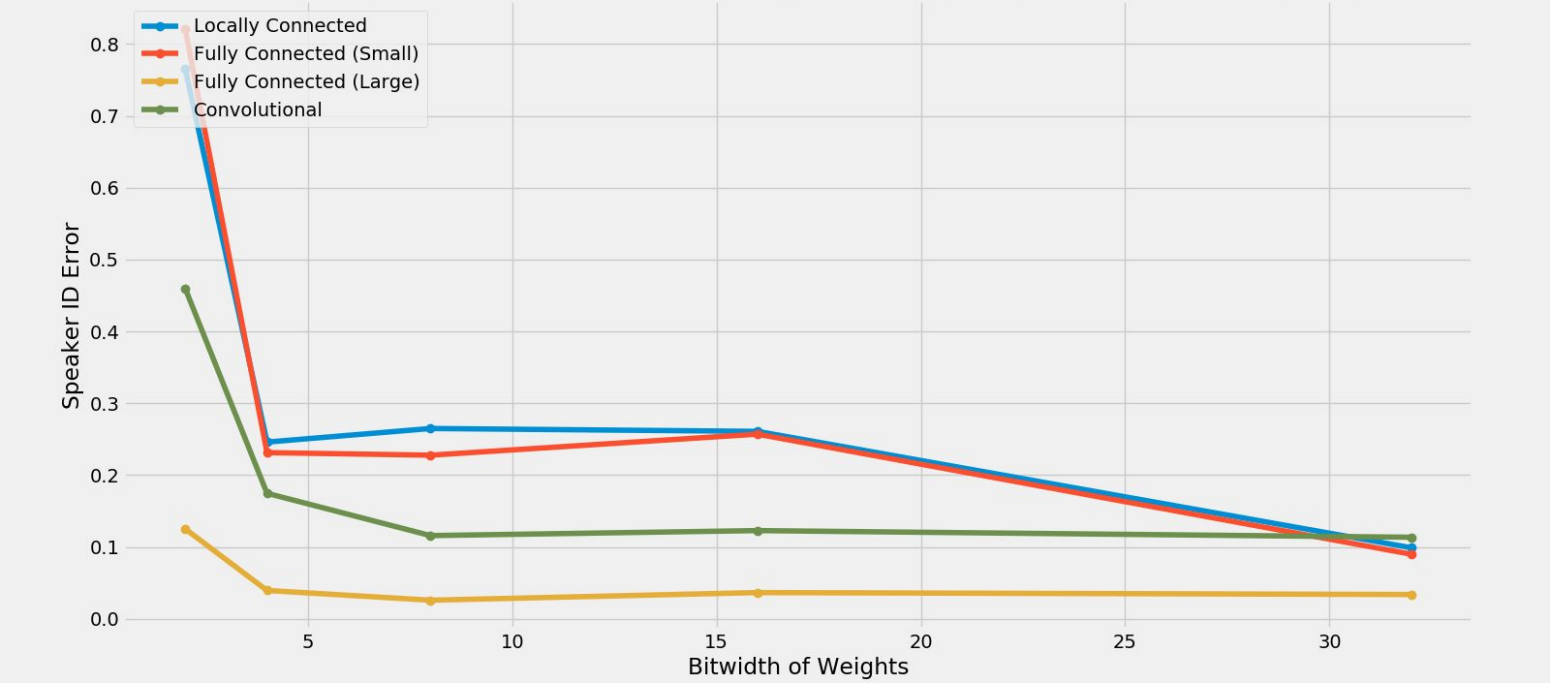
$$\mathtt{quant}(x) = \frac{1}{2^k} \ \mathtt{round}(2^k \times x)$$

- Activations are clipped to [0,1] using clipped ReLU and `quant(..)`
  - Low-bitwidth arithmetic!

Zhou, et al. 2016

Error/Size Trade-off with [-1,1] FxPt Quant.

Speaker ID Error vs. Bitwidth of Quantized Parameters



(Full Precision Activations)

[-1,1] FtPt Quant. Improves Footprint and Score

Model	Error	Mults	Params	Model Size	Score	Baseline
FCN, Large (4-bit)	0.039581	1399K	1399K	699.5KB	9.986	10.763
FCN, Large (8-bit)	0.026033	1399K	1399K	1399KB	10.287	10.763
FCN, Small (4-bit)	0.23138	519K	520K	260KB	9.892	10.404
FCN, Small (8-bit)	0.22782	519K	520K	520KB	10.186	10.404
CNN (4-bit)	0.1748	266K	260K	130KB	9.179	9.936
CNN (8-bit)	0.11593	266K	260K	260KB	9.301	9.936
LCN (4-bit)	0.26496	276K	287K	143.5KB	9.418	9.954
LCN (8-bit)	0.24604	276K	287K	287KB	9.688	9.954

FCN: 75% memory reduction, 8% reduction in p-score, no error increase

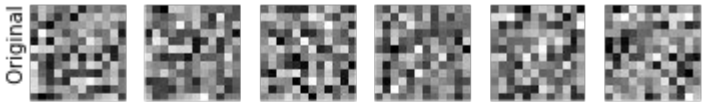
But Wait There's More: Trained Ternary Quant.

A different quantize op within the network's computation graph:

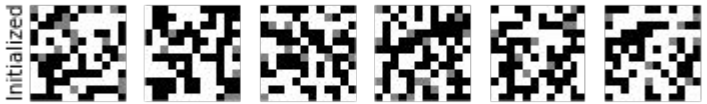
$$\widetilde{W}_{ijk}^{l} = \begin{cases} K_1^l & \text{if } W_{ijk}^l \geq \Delta^l \\ 0 & \text{if } -\Delta^l < W_{ijk}^l < \Delta^l \\ K_2^l & \text{if } W_{ijk}^l \leq -\Delta^l \end{cases}$$

K is ternarized value
Δ is ternarization threshold

Floating Point SID Kernels



Ternary SID Kernels



Activations are not quantized

Ternary allows lazy multiplications

Normal multiply-and-accumulate during feed-forward evaluation:

$$Wx = W_{11}x_1 + W_{12}x_2 \ldots$$

Ternary:

- ***W*** **is either** W_p **or** W_n
- **We accumulate-then-multiply for inference in our ternary system:**

$$Wx = W_p(x_1 + x_4 + \cdots + x_p) + W_n(x_2 + \cdots + x_q)$$

TTQ FCN: 100x Multiply, 8x Size Reduction

Model	Error	Mults	Params	Model Size	Score	Baseline
FCN, Large (Tern. Ends)	0.12884	6.56K	1399K	364.7KB	7.885	10.763
FCN, Large (Non-Tern. Ends)	0.11046	637.8K	1399K	2736KB	10.683	
CNN (Tern. Ends)	>0.99	-	-	-	-	9.936
CNN (Non-Tern. Ends)	0.38439	72.8K	286K	262KB	9.301	
LCN (Both)	>0.99	-	-	-	-	9.954
FCN, Small (Both)	>0.99	-	-	-	-	10.404

- Bootstrapped models from floating point
- Most models diverging, despite LR tuning and initialization
- For large FCN:
  - 27% reduction in metric score; multiplication excise was a win
  - Hit in speaker classification error

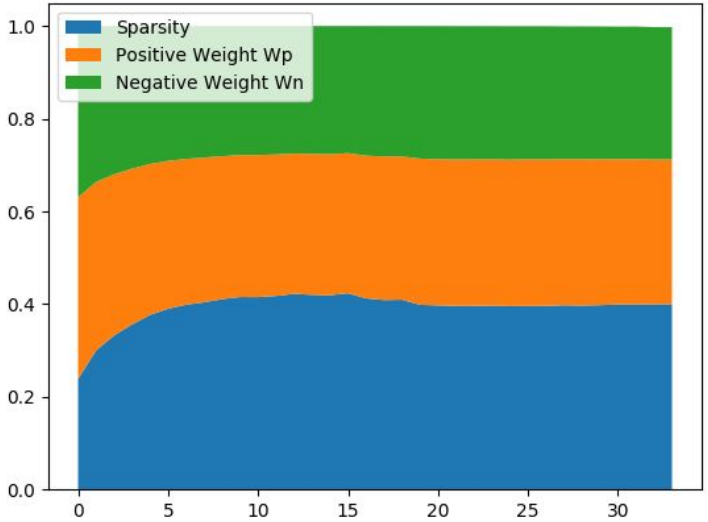
Running with Open Speaker Set and d-vectors

- Held-out set of speakers and the traditional d-vector/cosine distance
- Trial results match the trends witnessed using classification error: TTQ and 4-bit FxPt offer lowest metric scores

Model	Mults	Params	Bytesize	EER	EER-Score
FCN, Large (Full Prec.)	1399K	1399K	5596KB	7.03	13.740
FCN, Large (8-bit)	1399K	1399K	1399KB	7.94	13.191
FCN, Large (4-bit)	1399K	1399K	699.5KB	8.61	12.925
FCN, Large (TTQ)	6.56K	1399K	364.7KB	8.90	10.323
CNN (Full Prec.)	266K	260K	1044KB	8.05	12.349
CNN (8-bit)	266K	260K	261KB	9.44	11.816
CNN (4-bit)	266K	260K	130KB	9.81	11.530
CNN (TTQ, NT Ends)	72.8K	260K	262KB	20.82	11.598

Inducing Sparsity to Decrease Non-Zero Ops

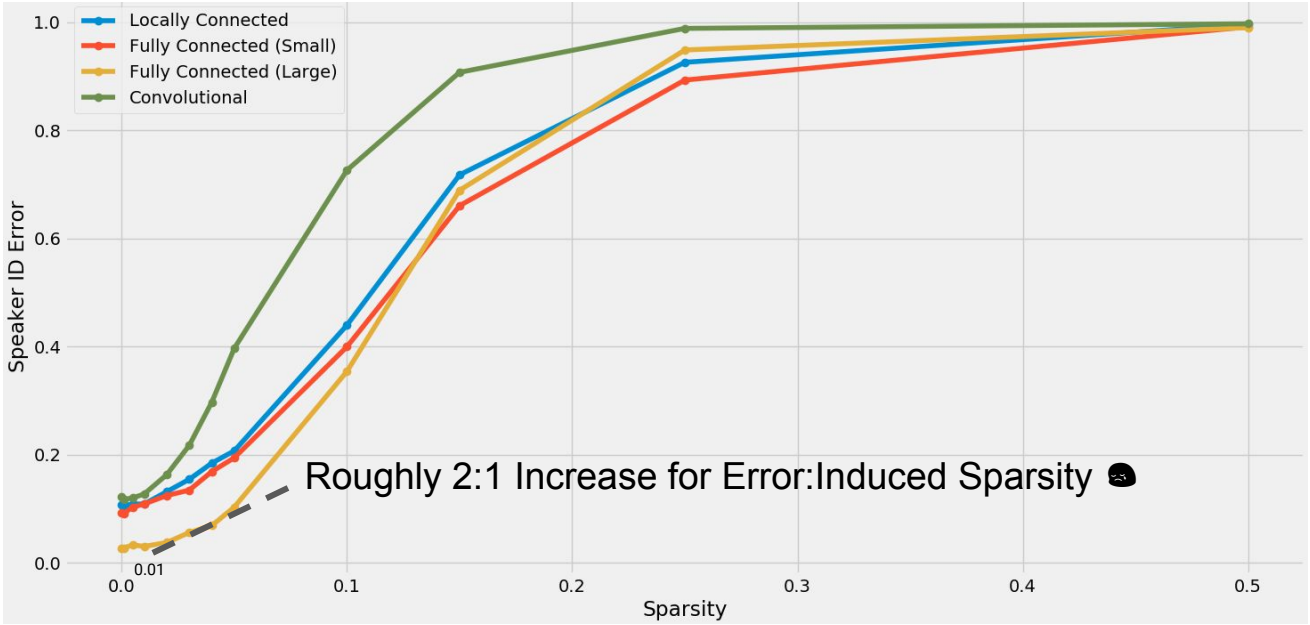
- Parameters in large ternary FCN are fairly sparse:



Sparsity decreases total op count, latency, and compressed storage footprint

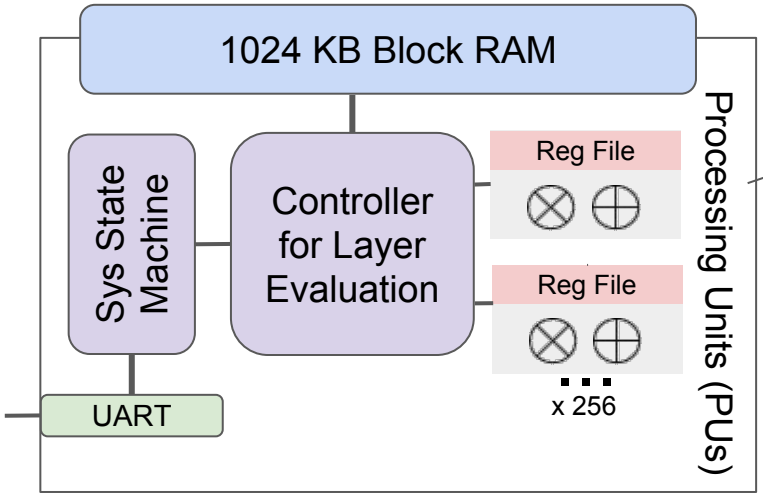
Inducing Sparsity to Decrease Non-Zero Ops

- Prune smallest weights to increase zero ops in 8-bit [-1,1]-FxPt models
- Twenty cycles of prune/retrain/prune/…



Translating to FPGA: Ternary Speaker ID

- FPGA system in progress:
  - Exploit parallelizable dot products
  - Scale bitwidth of hardware to match desired model precision: power/area savings
- Improved critical path latency, **lower-bound** power improvements of 10-30%

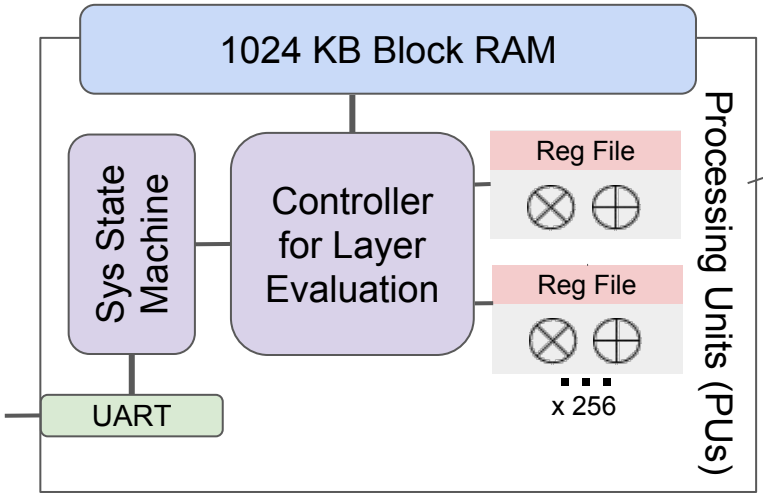


	PU Power (μW)	Slack (pS)
Ternary, 16bit Act.	100.142	3188
Ternary, 32-bit Act.	447.547	3130
Full, 16-bit Act.	158.451	1420
Full, 32-bit Act.	470.213	1376

100MHz clock, simulation using synthesis on LP 65nm CMOS

Translating to FPGA: Ternary Speaker ID

- FPGA system in progress:
  - Exploit parallelizable dot products
  - Scale bitwidth of hardware to match desired model precision: power/area savings
- Improved critical path latency, **lower-bound** power improvements of 10-30%



	4-bit FxPt, Energy, uJ	Full-Precision Energy, uJ
Small FCN Eval	1.235	2325
CNN Eval	1.228	2304
LCN Eval	1.838	2345

Energy reductions because of no off-chip memory access!
100MHz clock, simulation using synthesis on LP 65nm CMOS

Conclusions

- Speaker ID in memory-constrained embedded systems is feasible
- Achieve different points along the error/size trade-off applying:
  - Normalization Folding
  - [-1,1] Fixed Point Quantization
  - Trained Ternary Quantization (effective!)
  - Pruning
- Custom accelerators for speech tasks (VAD/Keyword/Speaker ID/ASR) offer power, area, and latency benefits