A Fast Network Exploration Strategy to Profile Low Energy Consumption for Keyword Spotting

Arnab Neelim Mazumder and Tinoosh Mohsenin
University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, USA

Introduction

- Keyword Spotting (KWS) relates to identification of Keywords in utterances ('Hey Siri', 'Ok Google', etc.).
- Voice assistant devices are power hungry and thus are a burden on battery life.
- However, KWS networks can act as a triggering mechanism for voice assistant devices to become active. In that way voice assistant devices do not have be always 'on'.
- However, KWS networks themselves need to be lightweight to enhance battery life.
- Making KWS networks lightweight require extensive parameter optimization.
- We propose a regression-oriented network exploration approach that leads to suitable parameter selection for efficient deployment on FPGA, resource constrained edge devices or commodity microcontrollers.

Problem Formulation

- The major factors that influence the power consumption of a deployed CNN can be listed as:
  - ❑ scaling of the filters,
  - ❑ quantization of the weights and data,
  - ❑ resolution of the inputs,
  - ❑ depth of the network,
  - ❑ sparsity of the network, and
  - ❑ interconnection between layers
- Usually, low precision (q) CNNs have increased number of filters (s) in each layer to compensate for accuracy loss.
- Similarly, CNNs with single precision floating point have reduced number of filters to compensate for high power consumption.
- There is a sweet-spot between quantization (q) and filter scaling (s) that leads to near optimum results for both accuracy and power consumption.
- We can make the following inferences for a deployed CNN:
  - ❖ $NN(size) \propto q.s^2$
  - ❖ $NN(largest\_fmap) \propto q.s$
  - ❖ $NN(computations) \propto s^2$
  - ❖ $NN(Mult\_Op\_Cost) \propto q^s.s^2$
  - ❖ $NN(Add\_Op\_Cost) \propto q.s^2$

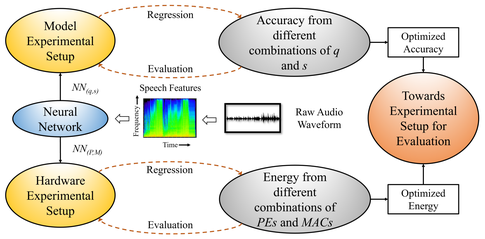

Figure:1 A high level overview of the required steps for experimental and analytical approach of our proposed methodology for keyword spotting.

- ❖ $minimize\ NN\ (Energy) for\ config \in \{P, M, q, s\}$ [1]
- This minimizes the energy generated for inference of these models provided that the target accuracy is maintained. Here, *P* and *M* correspond to the number of processing engines and multipliers of the hardware.

Dataset and Framework

- Google speech commands dataset [2] contains 30 different classes of audios.
- Audios have a duration ranging from 0.5s to 1s and contain keywords such as 'Yes', 'No', 'Up', 'Down', etc.
- Audios are converted to MFCC spectrogram using librosa library [3].
- Spectrogram input reduces computation for the CNN network.
- Audios are sampled at 22KHz with default values for MFCC parameters. Output spectrograms have a 44 x 13 shape.

Table 1: Framework used on the Google Speech Commands

Layer	Kernel Shape	#Filters	Stride
Conv2D	3 x 3	64*s*	1
Maxpool2D	2 x 2	64*s*	2
Conv2D	3 x 3	32*s*	1
Maxpool2D	2 x 2	32*s*	2
Conv2D	3 x 3	32*s*	1
Maxpool2D	2 x 2	32*s*	2
Fully Connected	-	64*s*	-
Fully Connected	-	#output	-
Total Computations (Millions)	$3.06\ s^2$		
Model Size (KB)	$4.20\ qs^2$		
Largest Feature Map (KB)	$3.70\ qs$		

Regression on Accuracy

- Train the CNN network for different configurations of *q* and *s*. For this KWS example, we train 12 different models where $q \in \{2,4,8\}$ & $s \in \{0.5,1,2,4\}$.
- It is thus possible to determine the accuracy results of the untrained combinations by running regression on the 12 data points and fitting the following rational polynomial:

$$Accuracy\ (NN(q,s)) \approx \frac{A_6.q.s + A_5.s + A_4.q + A_3}{q.s + A_2.s + A_1.q + A_0}$$

- The models are trained for 100 epochs with Adam optimizer.
- The learning rate reduces by 0.1 every 33 epochs.
- Quantization is performed with the Qkeras Library.
- We adopted the least square method for regression (RMSE=0.9).

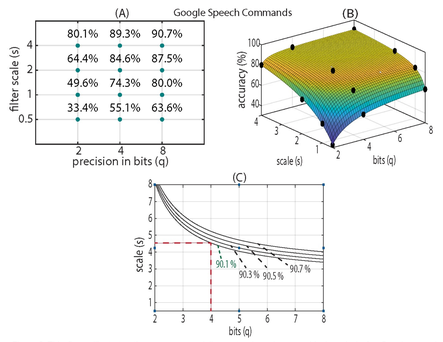

Figure:2 This figure illustrates the accuracy modeling setup based on empirical analysis for Google speech commands. (A) represents the accuracy experienced from training different *(q,s)* configurations, (B) illustrates the surface plot indicating the fit for the accuracy regression equation, and (C) demonstrates a better visualization of the relationship between accuracy, scaling, and precision through contours.

Table:2 Verification of Accuracy Regression with New Datapoints

(q,s)	Accuracy (%)	
	Actual	Pred.
(4.0,2.5)	86.5	86.4
(4.0,3.5)	88.7	88.3
(4.0,4.5)	**90.3**	**90.1**
(8.0,2.5)	88.7	88.5
(8.0,3.0)	89.6	89.4
(8.0,3.5)	90.2	90

Accelerator Design

- The design consists of four distinct parts:
  - Processing Engine (PE) Array - Address Generator
  - Maxpooling Block and - Memories
- Maxpooling layer uses bubble-sort strategy to sort the max value.
- The design can be extended to any number of PEs. Our implementations consider *16s* as PE value where *s* is the filter scaling. For simplicity of we use only 8 multipliers in each PE.

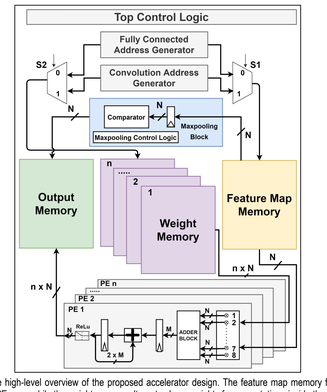

Figure:3 The high-level overview of the proposed accelerator design. The feature map memory forwards input data to the PE array while the weight memory alternates layer weights for computations inside the PE array. The output from the MAC operation is temporarily stored in the output memory. Along with this, the maxpooling block performs maxpooling function of the data with the help of a comparator where it bubble sorts the data to achieve proper feature map size. Top control logic regulates the state machine and pipelines the order of execution for convolution, maxpooling, and fully connected layers for precise hardware operation.

Regression on Energy

$$Energy\ (HW(q,s)) \approx (B_3.q^2.s^2 + B_2.q.s^2 + B_1.q.s + B_0)(D.s + E)$$

- The first and second term in the first parenthesis corresponds to multiplication and addition power. The third and fourth term relates to memory communication power and static power, respectively.
- The terms in the second parenthesis correspond to latency of the first and rest of the layers, respectively.
- The near-optimal configuration *(q=4.0, s=4.5)* is defined by the minima of the convex contour curve (red line).

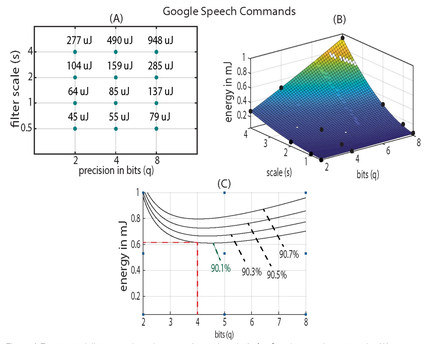

Figure:4 Energy modeling setup based on experimental analysis for Google speech commands. (A) represents the energy consumption experienced from implementing different configurations, (B) illustrates the surface plot indicating the fit for the energy equation, and (C) demonstrates a better visualization of the relationship between accuracy, energy, and precision through contours.

Table:3 Verification of Energy Regression with New Datapoints

(q,s)	*(P,M)*	Energy (mJ)	
		Actual	Pred.
(4.0,2.5)	(40,8)	0.21	0.23
(4.0,3.5)	(56,8)	0.39	0.41
(4.0,4.5)	**(72,8)**	**0.58**	**0.60**
(8.0,2.5)	(40,8)	0.41	0.42
(8.0,3.0)	(48,8)	0.56	0.59
(8.0,3.5)	(56,8)	0.74	0.76

Results and Comparison

- Our experiments were carried out on the Xilinx Artix-7 200t part for an operating frequency of 100 MHz.
- Artix-7 200t has 365 BRAMs (36Kb) = 1.64 MB of space.

Table:4 Implementation Results of Different Workloads on the Artix-7 200t FPGA at 100 MHz

(q,s)	*(P,M)*	BRAM	Pwr (W)	GOPJ	Latency (ms)
(4.0,1.0)	(16,8)	18	0.28	40.1	0.31
(4.0,2.0)	(32,8)	51	0.38	79.6	0.42
(4.0,4.0)	(64,8)	165	0.76	98.9	0.65
(4.0,4.5)	**(72,8)**	**185**	**0.83**	**104.9**	**0.7**
(8.0,1.0)	(16,8)	27	0.45	24.9	0.31
(8.0,2.0)	(32,8)	85	0.68	44.5	0.42
(8.0,4.0)	(64,8)	296	1.47	51.1	0.65

- We reduce the memory requirements by choosing a 4-bit implementation which also allows our design to have an energy and energy efficiency advantage of around 2.1x and 4x respectively compared to [4].

Table:5 Comparison of our *NN (q=4,s=4.5)* implementation on the XC7A200T platform to recent KWS hardware implementations.

Related Work	[4]	[5]	This Work	
Model	SCNN	CNN	CNN and FC	
Dataset	Speech Commands	Speech Commands	Speech Commands	
Precision	8-bits	8-bits	4-bits	
Accuracy (%)	88.1	87.6	90.1	
Device	XC7A200T	Cortex-M7 Microcontroller	XC7A200T	
Model Size (KB)	150	497	340	
Power (W)	1.04	-	0.47	0.83
Clock (MHz)	47.6	216	47.6	100
Latency (ms)	1.43	12	1.47	0.7
Energy (mJ)	1.49	-	**0.7**	0.58
GOPJ	10.5	-	**41.5**	104.9

Conclusion

- This setup allows a fast approach to obtain the near optimal configuration for desired accuracy and energy specifications for KWS.
- We can enhance the objective function with multiple objectives instead of focusing on single objective solution as a future direction.
- It is also possible to implement the setup with different combinations of network parameters in which case the problem might need a different solution as opposed to regression to get a near-optimal result.
- We can also use synthetic data from GANs to increase the number of datapoints for better curve fitting.

References

1. Morteza Hosseini, Mohammad Ebrahimabadi, Arnab Mazumder, Houman Homayoun, and Tinoosh Mohsenin. 2021. A Fast Method to Fine-tune Neural Networks for the Least Energy Consumption on FPGAs. In Proceedings of the Hardware Aware Efficient Training workshop of ICLR 2021.
2. Pete Warden. 2018. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. arXiv:1804.03209
3. Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In Proceedings of the 14th python in science conference, Vol. 8. Citeseer, 18–25
4. Gianmarco Dinelli, Gabriele Meoni, Emilio Rapuano, Gionata Benelli, and Luca Fanucci. 2019. An fpga-based hardware accelerator for cnns using on-chip memories only: Design and benchmarking with intel movidius neural compute stick. International Journal of Reconfigurable Computing 2019 (2019)
5. Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra. 2017. Hello edge: Keyword spotting on microcontrollers. arXiv preprint arXiv:1711.07128 (2017).