AI Voice Processing on an FPGA

Andrew Rooney, Nathan Mikhail, Justin Diala, Ramana Vasanthan

ECE 492
Client/Technical Advisor: Dr. Steven Knudsen
Year: 2022

Background

We sought to develop a low latency noise filtering and keyword detection solution using a machine learning (ML) algorithm on an edge device. The interest in ML for audio filtering stems from its potential capacity to outperform the current state of the art in audio filtering at filtering stationary, non-stationary, and babble noise in an adaptive manner [1].

Many ML algorithms are computationally expensive and are not optimized for edge deployment. Real time adaptive filtering would be useful for defense applications, where personnel often find themselves in loud environments where communication is vital and invoking actions with keywords is practical.

Results

The RNN filter runs on our system with a total of 27ms of latency. Latency is dominated by software buffering as shown in Table 1 but is capable of real time operation.

The filter performed very well with stationary and babble noise, capably passing the primary speaker and supressing noise. This is seen clearly in Figures 1 and 2. Figure 1 shows an audio sample with a primary speaker mixed with a din of background chatter (babble noise). Figure 2 shows the same audio filtered with the RNN model.

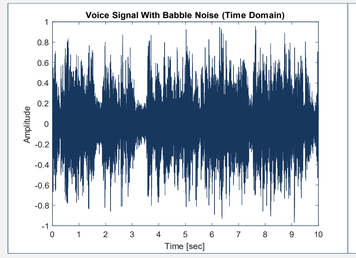


Figure 1: Babble Noise Before filtering

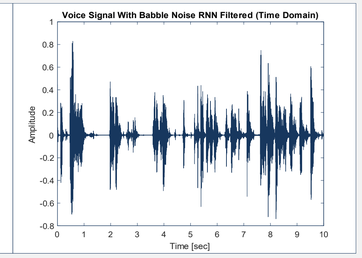


Figure 2: Babble Noise After filtering

	Latency (ms)
Preprocessing	0.3 (/10ms of audio)
Postprocessing	0.1 (/10ms of audio)
RNN filter inference	0.3 (/10ms of audio)
Total audio latency (includes software buffering)	27

Table 1: System Latency

	White Noise (stationary noise)	Dog Barking (non-stationary noise)	Babble Noise
Unfiltered SNR (dB)	3.5004	4.4527	6.5315
Filtered SNR (dB)	48.4019	4.0554	36.7023

Table 2: Signal to Noise Ratio Measurements

Table 2 shows the signal to noise ratio of audio samples before and after filtering with various noise sources. The RNN filtering model did not perform well with non-stationary noise. There were also minor auditory artifacts of the software processing.

Our RNN model, when targeted for ZU3EG Ultrascale+ MPSoC at 300MHz, initially utilized floating-point arithmetic, resulting in 213% Lookup-Table (LUT) utilization and 3.8ms of latency; this was improved by using (22,14) fixed-point arithmetic. This, along with various other optimizations yielded 0.25ms of latency (which largely agrees with the results in Table 1) and 67% LUT utilization. Once the design was fully implemented, LUT utilization decreased again to 48%.

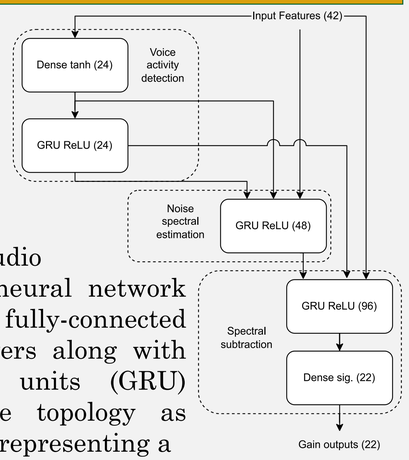
Our keyword detection model was estimated to have a 50ms latency when running on an offline PC.

Solution

Our solution uses a Xilinx Ultrascale+ multi-processor system on a chip (MPSoC) to accelerate an ML model on a field programmable gate array (FPGA) in real time. Two machine learning architectures were investigated – a convolutional neural network (CNN) and a recursive neural network (RNN). We found that a RNN based implementation was best suited to our project's requirements [2].

Software running on the SoC's ARM Cortex-A53 processing system (PS) uses the FreeRTOS operating system to buffer incoming/outgoing audio data, perform audio pre-/post-processing and execute RNN inference on the FPGA.

Google's "Teachable Machine" was used to create a keyword detection ML model and was implemented on an offline PC.



The RNN's recurrent connections give it the ability to model time sequences; this makes it an ideal architecture for audio processing. Our neural network (NN) uses dense, fully-connected non-recurrent layers along with gated recurrent units (GRU) connected in the topology as shown – each box representing a layer of neurons. *Tanh*, *ReLU* and *sigmoid* activation functions are used to determine when a node will trigger. Our NN accepts pre-processed features and returns a set of gains, used to attenuate spectral bands in the original audio signal. [2]

Discussion

We successfully created a system that demonstrates real time noise suppression using ML algorithms on an MPSoC, using the FPGA for acceleration, and showed how keyword detection could be done on an edge device.

However, the system displayed several quirks during operation; for instance, the NN takes a few moments to understand the input and learn how to properly supress noise, which is why we see poor performance with non-stationary noise.

The next steps to begin to refine the proposed system include:

- Further research of RNN implementations
- Resource Optimization
- Refinement of audio data pipeline
- Keyword model integration
- Additional model training

References

[1] P. Guduguntla, "Background noise removal: Traditional vs AI algorithms," *Medium*, 18-Mar-2021. [Online]. [Accessed: 29-Mar-2022].

[2] J.-M. Valin, "A hybrid DSP/deep learning approach to real-time full-band speech enhancement," *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, 2018.

We would like to thank Dr. Bruce Cockburn for lending us his Xilinx Ultrascale+ development board, and Dr. Steven Knudsen for his valuable advice.