BinaryCmd: Keyword Spotting with deterministic binary basis

Javier Fernández-Marqués[†], Vincent W.-S. Tseng[‡], Sourav Bhattachara[*], Nicholas D. Lane[†,*]

[†] University of Oxford, [‡] Cornell University, [*]Nokia Bell Labs

ABSTRACT

We present a binary architecture with 60% fewer parameters and 50% fewer operations during inference compared to the current state of the art for keyword spotting (KWS) applications at the cost of 3.4% accuracy drop.

1 A BINARY NETWORK FOR KWS

KWS has become a popular always-on feature in smartphones, wearables and smart home devices. It serves as the entry point for speech based applications once a predefined command (e.g. "Ok Google", "Hey Siri") is detected from a continuous stream of audio. Because KWS applications are alwasy running they follow a very eficient architectural design and are often implemented on small dedicated microcontrollers. These devices are contrained in terms of memory and compute capabilities, limiting the complexity and memory footprint of the deployed model.

We compare our work to *HelloEdge* [3] following their microcontroller classification scheme and particularly focusing on the Small (S) group, where the model size limited to 80kB and the maximum number or OPs during inference is 6M. Likewise, we use Google's Speech Commands Dataset [2] to evaluate our architecture.

System Overview. The implemented KWS system is comprised of two fundamental blocks where speech features are first extracted from the 1*s* voice command input and are fed to a NN-based block that outputs the id of the detected voice command. The system's macroarchitecture is depicted in Figure 1. We follow the same strategy as in [3] to extract an array of 49×10 MFCC speech features from the input speech signal and feed them to our network.

Architecture. We present a novel NN block containing the following elements: three nested *on-the-fly* convolutional layers (Figure 2) followed by a standard convolutional, max-pooling and fully connected layers.

***On-the-fly* convolutions**. Unlike standard convolutional neural networks (CNN), our architecture learns weighting coefficients of deterministic binary basis that are combined in a linear fashion manner to generated the filters. We use orthogonal variable spreading factor codes[1] of length 2^n, $n \in \mathbb{N}$, to generate these basis.

2 EVALUATION

We evaluate three configurations of BinaryCmd with a focus on reducing on-device memory footprint and number of OPs per inference pass.The three configurations only differ in the number of filters, stride and ratio parameters used in our on-the-fly convolutional layers. Intuitively, the smaller the ratio, the coarser the filters and the bigger the model size savings would be, and vice-versa.

We compare BinaryCmd against DS-CNN and all the baselines analysed in [3]. The preliminary results (Figure 3) show the potential of our binary architecture: up to 60% model size and 67% number of OPs reduction at the expense of no more than 3.4% accuracy loss when compared to DS-CNN. We have applied standard 8-bit quantisation to the majority of the layers, meaning that further optimisation is possible. All three of our configurations simultaneously achieve top *accuracy-to-size* (A2S) and *accuracy-to-OPs* (A2OPs) ratios meaning that BinaryCmd is a good first step towards the design of architecture capable of providing over 90% accuracy levels with minimal memory footprint and low computational costs.

[1]OVSF codes were introduced for 3G communication systems as channelizations codes aiming to increase system capacity in multi-user access scenarios [1].

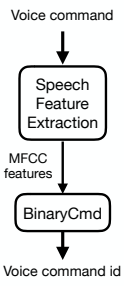

Figure 1: System architecture.

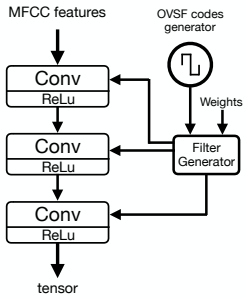

Figure 2: BinaryCmd's core.

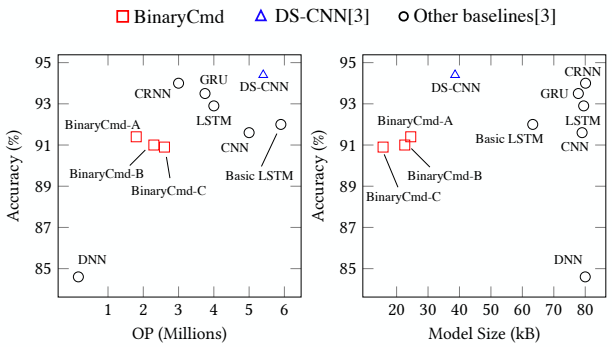

Figure 3: Results comparison against architectures in [3] for the category of small (S) microcontrollers limited to 6M OPs and 80kB.

REFERENCES

[1] Fumiyuki Adachi, Mamoru Sawahashi, and Hirohito Suda. 1998. Wideband DS-CDMA for next-generation mobile communications systems. *IEEE communications Magazine* 36, 9 (1998), 56–69.

[2] Pete Warden. 2017. Speech Commands: A public dataset for single-word speech recognition. (2017). http://download.tensorflow.org/data/speech_commands_v0.01.tar.gz

[3] Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra. 2017. Hello Edge: Keyword Spotting on Microcontrollers. *CoRR* abs/1711.07128 (2017). arXiv:1711.07128