On-device Chatbot System using SuperChat Method on Raspberry Pi and CNN Domain Specific Accelerator

Hao Sha
Gyrfalcon Technology Inc.
Milpitas, CA

Baohua Sun
Gyrfalcon Technology Inc.
Milpitas, CA
baohua.sun@gyrfalcontech.com

Nicholas Yi
Gyrfalcon Technology Inc.
Milpitas, CA

Wenhan Zhang
Gyrfalcon Technology Inc.
Milpitas, CA

Lin Yang
Gyrfalcon Technology Inc.
Milpitas, CA

Abstract

Chatbot is a popular interactive entertainment device requires semantic understanding and natural language processing of input inquiries and appropriate individualized responses. Currently, most chatbot services are provided with connection to cloud due to the limitation of computation power on edge devices, which brings in the privacy and latency concerns. However, the recent research on SuperChat method shows that the chit- chat tasks can be solved using two-dimensional CNN models. In addition, low-power CNN Domain Specific Accelerators are in wide availability since the past two or three years. In this paper, we implement SuperChat method on a Raspberry Pi 3.0 connected through USB to a low-power CNN accelerator chip, which is loaded with the quantized weights two-dimensional CNN model. The resulting system can reach convincing accuracy with high power, memory efficiency, and very low power consumption.

1 Introduction

Chatbots such as Apples Siri and Amazons Echo are widely used today to interactively carry out simple tasks and answer questions. These chatbots use cloud computing technology for both semantic understanding and natural language processing of input inquiries and appropriate individualized response. However, this comes at a cost of offline unavailability and privacy concerns with human voice data being communicated and stored. In addition, the desire arises for these chatbots to emulate human characteristics in personalized behavior and response. There also arises a need for localized chatbot solutions in dealing with specific areas, such as senior centers, kids toys, etc.

The SuperChat solution [8] is applied to solve the above problems. It uses the two-dimensional embedding of the state-of- art Super Characters method [7] for text classification operations to achieve high quality, engaging responses. Super Characters method is also extended to tabular data machine learning [1], image captioning [4], and Multi-Modal sentiment analysis [5]. Low-power CNN accelerators are wide available to implement the CNN

models in these methods. Sun, et al. (2017) has designed a Convolutional Neural Networks Domain Specific Architecture (CNN-DSA) accelerator for extracting features out of an input image [9, 6]. It processes 224x224 RGB images at 140fps with ultra-power-efficiency, a record of 9.3 TOPS/Watt and peak power less than 300mW. Super Characters deployed on these low-power devices [3] shows the promissing availability on edge devices.

In this paper, we propose a low-cost solution for chatbot, where the core SuperChat engine is all localized.

2 Related Work

2.1 SuperChat

Figure 1 illustrates the SuperChat method that was used in our system. The response sentence is predicted sequentially by predicting the next response word in multiple iterations. During each iteration, the input sentence and the current partial response sentence are embedded into an image through two-dimentional embedding. The resulting image is called as a SuperChat image. And then this SuperChat image is fed into a CNN model to predict the next response word. In each SuperChat image, the upper portion corresponses to the input sentence, and the lower portion corresponses to the partial response sentence. At the beggining of the iteration, the partial response sentence is initiallized as null. The prediction of the first response word is based on the SuperChat image with only the input sentence embedded, and then the predicted word is added to the current partial response sentence. This iteration continues until End Of Sentence (EOS) appeared. Then, the final output would be a concatenation of the sequential output.

Although the examples used in Figure 1 is illustrated with Chinese sentences, however, it can be also applied to other languages. For example, Asian languages such as Japanese and Korean, which has the same square shaped characters as in Chinese. For Latin languages where words may have variant length, SEW method [2] could be used to convert the Latin languages also into the squared shape before applying the SuperChat method to generate the dialogue response.

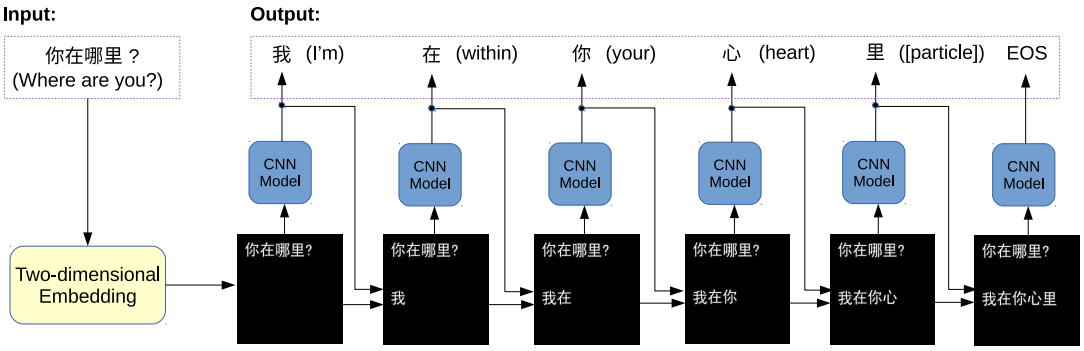

Figure 1: Super Chat.

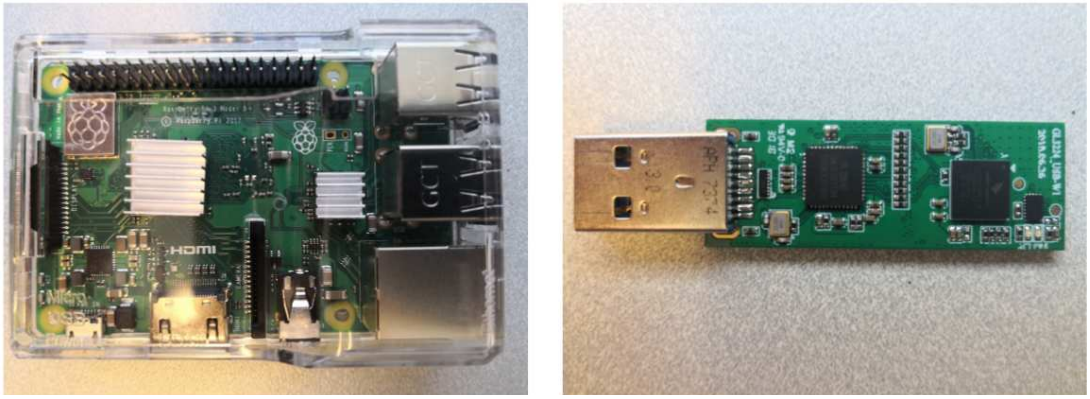

Figure 2: Raspberry Pi and GTI 2801 dongle

3 Chatbots Implementation

In order to implement Super Chat method, the system uses a low-cost Raspberry Pi single board computer to perform voice recording, audio playback, Internet/Cloud accessing, etc. and uses Gyrfalcons Edge-Computing Device (GTI 2801 dongle) to perform sentiment analysis as well as generate appropriate response, as shown in Figure 2. Cloud servers are temporarily used to perform Speech- to-Text and optionally Text-to-Speech. While the Speech-to- Text and Text-to-Speech module can be easily replaced by licensed services at this moment, we were temporarily using cloud-based free services to prove the concept for sake of convenience.

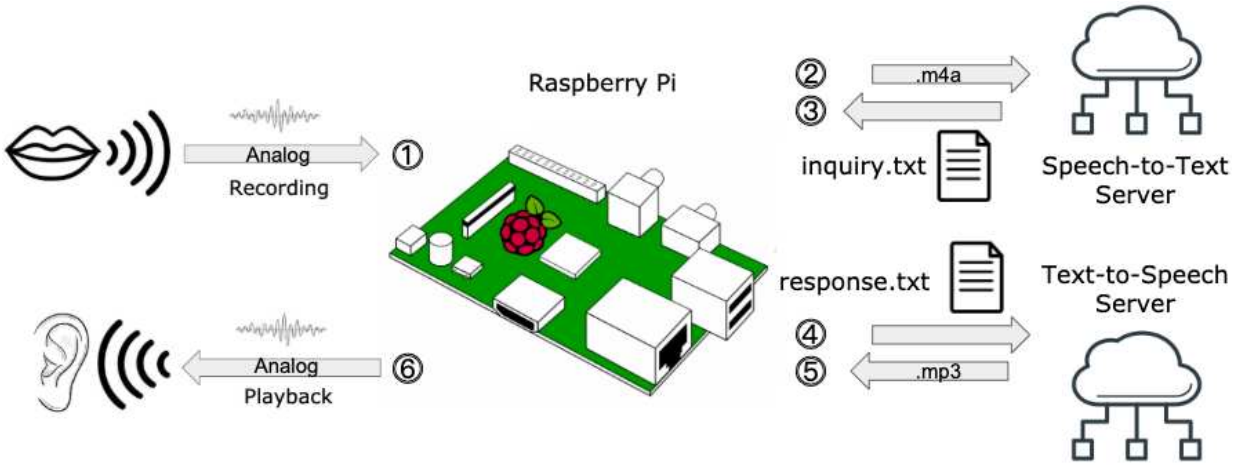

Figure 3: System Diagram.

3.1 System Structure

Figure 3 illustrates the full proposed system structure of the proposed SuperChat implementation, which will be discussed in details later. Inquiry text input may be received via two ways: transcribed audio to text (steps 1-3) or direct text from keyboard (step 3).

3.2 Voice Recording and Playback

Recording software, FFmpeg, which is a free open-source project, is used to capture and compress the voice to AAC format for better quality and compression ratio. AMR audio format can also be used, but is not supported by FFmpeg on Raspberry Pi. Ffmpeg is also used for playback.

3.3 Speech Recognition (Speech-to-Text) and Natural Language Processing

Using the Baidu Speech Recognition API, audio is recorded in AAC (m4a) format, which has a high compression rate. Baidu offers speech-to-text and text-to-speech via the cloud, and both operations average 2 seconds for sentences of 10 characters or shorter, with the majority of time spent transmitting and receiving the audio file. The Tencent Speech Recognition API is also successful in performing Speech Recognition and transcription, but it does not support AAC compression audio format, so the default WAV file takes a much longer time (10 seconds) as it is much larger. Before resolving to use AAC and M4A audio formats, WAV and AMR formats were tried. WAV is supported on initial tests with a laptop, but due to the low compression rates, yielded a large delay when reading and extracting speech in the cloud. AMR has a similar compression rate to AAC and M4A, with its speech-to-text being around 5 times faster than WAVs, but due to AMR format not being supported by Raspberry Pi, it was ultimately substituted out in favor of AAC.

Although the current solution involves the cloud-based s2t, while this module could be easily replaced with licensed softwares. A possible way for the on-device chatbot to be independent to the cloud service for this s2t module will be to purchase license from third parties. For example, iFlyTech has local translation device which first recognize the voice as text, and then translate text into a target language. The voice recognition module could be implemented on device, and the entire on-device translator product is less than $200 which means the voice recognition module could be implemented on-device with affordable cost.

3.4 On Chip SuperChat Engine

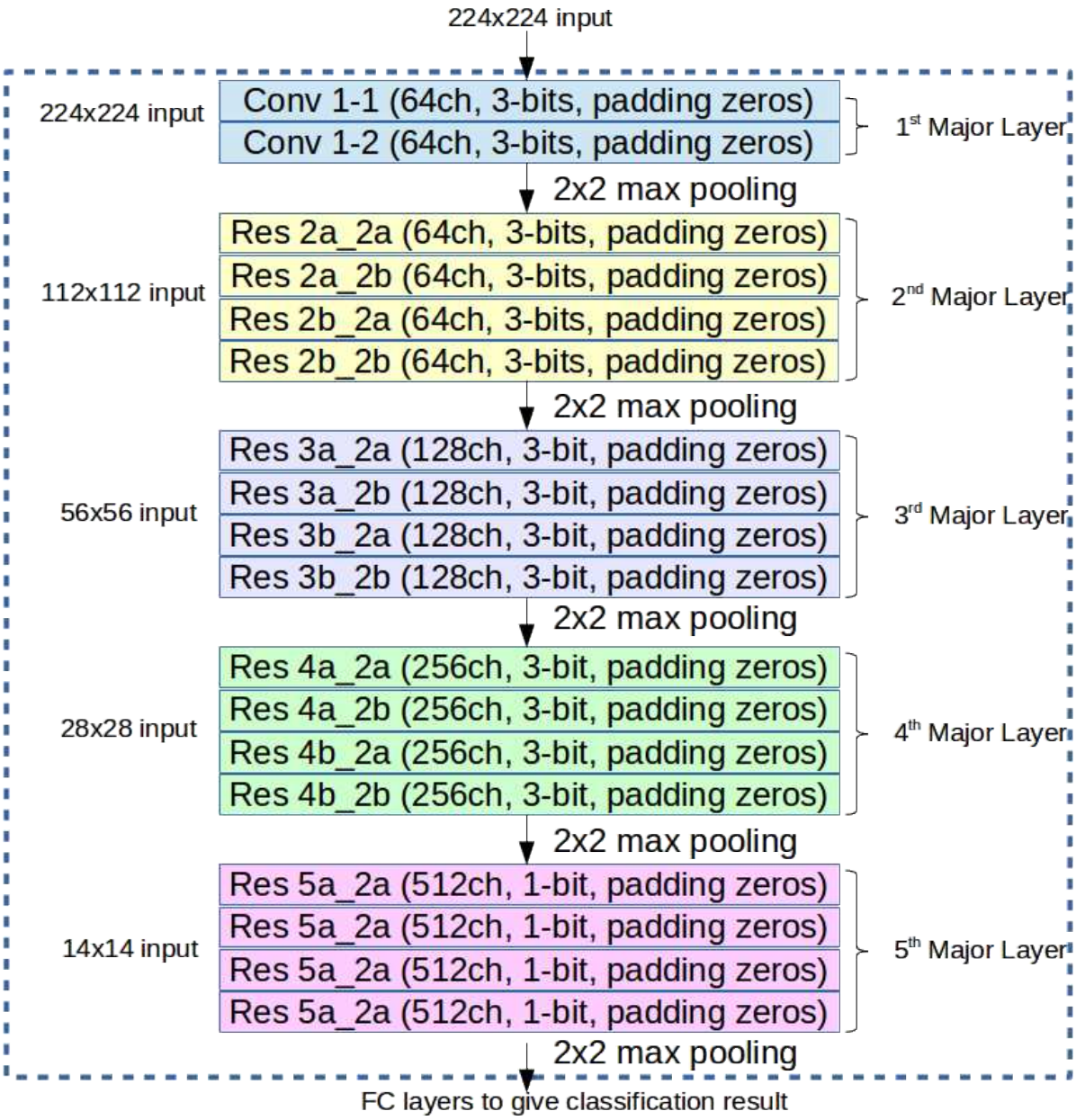

Figure 4: Gnet18 Model Architecture and Quantized Weight.

A trained SuperChat model is stored locally on the Raspberry Pi and loaded to a GTI 2801 dongle. The model being used is a quantinized Gnet18 model, as showed as Figure 4, which is a modified ResNet 18 model with all shortcut removed. The first four major layers uses 3-bits precision and the last major layer uses 1-bit precision. All activations are presented by 5-bits in order to save on-chip data memory. The representation mechanism inside theaccelerator supports up to four times compression with the 1-bit precision, and two times compression with the 3-bits precision. To efficiently use the on-chip memory, the model coefficients from the fifth major layers are only using 1-bit precision. For the first four major layers, 3-bits model coefficients are used as fine- grained filters from the original input image.

After CNN layers, FC layers are implemented on CPU before output prediction. The calculation power required by FC layer is negligible.

The CNN-DSA chip processing time is 15ms, and the pre- processing time on mobile device is about 6ms. The time for FC layer is 1 ms, and post-processing is negligible, so the total text classification time is 22ms. It can process nearly 50 sentences in one second, which satisfies more than real-time requirement for NLP applications like chatbot. [10]

3.5 Speech Synthesis (Text-to-Speech)

The response TXT file is sent to a separate Baidu Cloud Text- to-Speech server. The server will send back an audio file that was selected to be MP3 format in order to save the communication bandwidth.

To use offline text-to-speech, within the Python library there is an open source software speech synthesizer called eSpeak, which uses a formant synthesis method. It has many languages in a compact, small package. Ekho is another option for offline Text-to-Speech, which can be found at [11].

4 Results

Initial tests on a Raspberry Pi (ARM8) proved successful, with nearly perfect accuracy for audio transcription with varying degrees of loudness/length input. Output could be easily changed to accommodate volume, speed, speaker voice, etc..

5 Future Work

Ideally, offline Speech-to-Text can be used to implement a fully localized chatbot solution. The existing solutions are only found on Android devices for simple commands..

References

[1] Sun, Baohua, Lin Yang, Wenhan Zhang, Michael Lin, Patrick Dong, Charles Young, and Jason Dong. "Supertml: Two-dimensional word embedding for the precognition on structured tabular data." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 0-0. 2019.

[2] B. Sun, L. Yang, C. Chi, W. Zhang, and M. Lin. Squared english word: A method of generating glyph to use super characters for sentiment analysis. arXiv preprint arXiv:1902.02160, 2019.

[3] Sun, Baohua, Lin Yang, Michael Lin, Wenhan Zhang, Patrick Dong, Charles Young, and Jason Dong. "System Demo for Transfer Learning across Vision and Text using Domain Specific CNN Accelerator for On-Device NLP Applications." arXiv preprint arXiv:1906.01145 (2019).

[4] Sun, Baohua, Lin Yang, Michael Lin, Charles Young, Patrick Dong, Wenhan Zhang, and Jason Dong. "Supercaptioning: Image captioning using two-dimensional word embedding." arXiv preprint arXiv:1905.10515 (2019).

[5] Sun, Baohua, et al. "Multi-modal Sentiment Analysis using Super Characters Method on Low-power CNN Accelerator Device." arXiv preprint arXiv:2001.10179 (2020).

[6] Baohua Sun, Daniel Liu, Leo Yu, Jay Li, Helen Liu, Wenhan Zhang, and Terry Torng. 2018. MRAM Co-designed Processing-in-Memory CNN Accelerator for Mobile and IoT Applications. arXiv preprint arXiv:1811.12179 (2018).

[7] Baohua Sun, Lin Yang, Patrick Dong, Wenhan Zhang, Jason Dong, and Charles Young. 2018. Super Characters: A Conversion from Sentiment Classification to Image Classification. In Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. 309315. (2018)

[8] Sun, Baohua, et al. "SuperChat: dialogue generation by transfer learning from vision to language using two-dimensional word embedding." Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data. 2019.

[9] Baohua Sun, Lin Yang, Patrick Dong, Wenhan Zhang, Jason Dong, and Charles Young. 2018. Ultra Power-Efficient CNN Domain Specific Accelerator with 9.3 TOPS/Watt for Mobile and Embedded Applications. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 16771685. (2018)

[10] B. Sun, L. Yang, M. Lin, W. Zhang, P. Dong, C. Young, J. Dong, System Demo for Transfer Learning across Vision and Text using Domain Specific CNN Accelerator for On-Device NLP Applications, arXiv:1906.01145.

[11] https://www.eguidedog.net/ekho.php