PROJECT PROPOSAL: HOW DO YOU TURN THIS ON ON-DEVICE AUTOMATIC SPEECH RECOGNITION

Yi-Ting Yeh and Ting-Rui Chiang
Carnegie Mellon University
{yitingye,tingruic}@cs.cmu.edu

1 MOTIVATION

Automatic speech recognition (ASR) has been a key component in different applications such as intelligent assistants. To facilitate on-device applications, deploying ASR to edge devices is important. The ASR system needs to be run offline when no internet access is available. Having an offline ASR also preserves users' privacy since we no longer need to send user's private information to cloud servers. Therefore, in this project, we want to study how to perform ASR on the edge device.

2 HYPOTHESES AND APPROACHES

End-to-end ASR models are suitable for edge devices since we can easily accelerate them by distillation, pruning, and quantization techniques. However, simply deploying ASR models on the edge device might not be a trivial task even though the library supports edge computing. It is because edge devices have different hardware specifications. For example, the edge device we will use, Raspberry Pi 4, does not have a matrix computation accelerator. The embedded environment can also cause some challenges for executing the library. We must spend lots of effort in order to deploy and run an ASR model with reasonable latency.

We will aim to adapt a CTC-based model (Graves et al., 2006) with the library ESPNet 2 (Watanabe et al., 2018; Inaguma et al., 2020; Hayashi et al., 2020; Li et al., 2021) to Raspberry. A CTC-based model does not have separated encoder and decoder components. We conjecture that it will be the most efficient model architecture. The first acceleration method we will try will be layer and head pruning (Lee et al., 2021), because we suppose it will bring the greatest speedup. Depending on the performance difference between a CTC-based model and a transformer-based, knowledge distillation from a transformer-based model to a CTC-based model may be a viable direction. We may also try model quantization. However, we expect that smaller memory footprint will be the only benefit. We do not expect it will accelerate the model, since the CPU on Raspberry Pi 4 is not specialized for Int8 operations. We may also try other CTC variants, such as mask CTC (Higuchi et al., 2021) that utilizes a mask language model to refine the output. However, this may require implementing some new functions in the ESPNet library, and thus may take much more time.

3 EXPERIMENTAL SETTINGS

Experimental Design We will first use 100 hour subset of LibriSpeech (Panayotov et al., 2015) to run experiments. The word error rate (WER) will be used to evaluate basic performance of ASR models. Latency, FLOPS, and power consumption are used as metrics to evaluate model efficiency. Seq2seq model (Chan et al., 2016), attention model (Chorowski et al., 2015), Transducer (Graves, 2012), and CTC-based model (Graves et al., 2006) are considered as our ASR baselines. Transformers-based models (Karita et al., 2019) and the knowledge distillation will also be explored in this project as described in Section 2.

I/O The input will be raw wav file, and the output will be raw text. We will use the library `sounddevice` for recording.

Hardware

- A better microphone. Although modern ASR models will use data augmentation to make model more robust to noise, usually a better microphone will lead to better performance.
- A larger and faster SD card. We expect we will compare many models on the device. Having faster SD card with larger storage size will facilitate our experiments.

Off-device Training We will need to do off-device training in order to experiment on various acceleration methods such as knowledge distillation. We might also want to experiment on different model architecture whose pretrained parameters are not available. Therefore, It would be best to have about 200 dollars AWS credits to do off-device training.

4 Related Work

End-to-end ASR systems have shown great success thanks to the recent advance of deep learning technologies. CTC loss (Graves et al., 2006), attention (Chorowski et al., 2015), and their joint models (Kim et al., 2017; Hori et al., 2017) are strong models which could be used as baselines in our projects. Recently, Transformer-based ASR models (Karita et al., 2019) show very strong performance, but how could we deploy it on the end device is not well explored. Therefore, the knowledge distillation from Transformer to rnn-based models will also be studied in this project. Google researchers showed Transducer (Graves, 2012) architecture could be particularly useful in on-device speech recognition (He et al., 2019) especially for Pixel phones, and recent variants of Transducer also achieved very strong performance on LibriSpeech (Zhang et al., 2020). It will be interesting to do a comparison between CTC models and Transducer models on the Raspberry Pi.

5 Potential Challenges

Training time : The training time may be very long. To adapt to this challenge, we may use a pretrained model, and use only part of the training data for distillation.

Mismatch target distribution : The audio collected by the microphone may be very different from the audio in the training data. We may have to focus on the performance on the standard testing dataset. Or we may have to make the model more robust to distribution shift due to the microphone. Possible solutions include data augmentation or adversarial training.

Ethical Concerns : Other people besides the user should know when the ASR system starts to record the speech since they might not want their speech to be recorded. Thus, it is better to have an indication on the device indicating when the device starts.

6 Potential Extensions

One potential extension would be making the model real-time. However, this will require a totally different model, and ESPNet may not support it. We expect that this will require a great amount of efforts.

Another potential extension would be execting a text-to-speech (TTS) model. It is possible since ESPNet also support TTS models. It would be interesting to see whether the methods we proposed are applicable to TTS models.

We can also explore how to deal with the cocktail party problem in on-device setting. If we want to handle this problem, we might need to build a microphone array.

7 Timeline and milestones

- Oct 7: Solve problems and successfully run ESPNet 2 on Raspberry Pi 4.
- Oct 14: Collect the performance (speed/WER) of baseline models.
- Oct 28: Finish head and layer pruning on the CTC model respectively.

- Nov 11: Combine head and layer pruning.
- Nov 18: Quantize the model (tentative).
- Dec 2: Implement Mask-CTC (tentative).
- Dec 9: Accelerate Mask-CTC (tentative).

ACKNOWLEDGEMENT

We thank Prof. Shinji Wanatabe for providing us many suggestions and references.

REFERENCES

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4960–4964, 2016. doi: 10.1109/ICASSP.2016.7472621.

Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. *arXiv preprint arXiv:1506.07503*, 2015.

Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376, 2006.

Tomoki Hayashi, Ryuichi Yamamoto, Katsuki Inoue, Takenori Yoshimura, Shinji Watanabe, Tomoki Toda, Kazuya Takeda, Yu Zhang, and Xu Tan. Espnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7654–7658. IEEE, 2020.

Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al. Streaming end-to-end speech recognition for mobile devices. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6381–6385. IEEE, 2019.

Yosuke Higuchi, Nanxin Chen, Yuya Fujita, Hirofumi Inaguma, Tatsuya Komatsu, Jaesong Lee, Jumon Nozaki, Tianzi Wang, and Shinji Watanabe. A comparative study on non-autoregressive modeling for speech-to-text generation. 2021.

Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan. Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm. *arXiv preprint arXiv:1706.02737*, 2017.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. ESPnet-ST: All-in-one speech translation toolkit. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 302–311, Online, July 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.acl-demos.34`.

Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al. A comparative study on transformer vs rnn in speech applications. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 449–456. IEEE, 2019.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4835–4839. IEEE, 2017.

Jaesong Lee, Jingu Kang, and Shinji Watanabe. Layer pruning on demand with intermediate ctc. *arXiv preprint arXiv:2106.09216*, 2021.

Chenda Li, Jing Shi, Wangyou Zhang, Aswin Shanmugam Subramanian, Xuankai Chang, Naoyuki Kamo, Moto Hira, Tomoki Hayashi, Christoph Boeddeker, Zhuo Chen, and Shinji Watanabe. ESPnet-SE: End-to-end speech enhancement and separation toolkit designed for ASR integration. In *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, pp. 785–792. IEEE, 2021.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pp. 2207–2211, 2018. doi: 10.21437/Interspeech.2018-1456. URL http://dx.doi.org/10.21437/Interspeech.2018-1456.

Yu Zhang, James Qin, Daniel S Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V Le, and Yonghui Wu. Pushing the limits of semi-supervised learning for automatic speech recognition. *arXiv preprint arXiv:2010.10504*, 2020.