Robust Speech Recognition Model Using Multi-Source Federal Learning after Distillation and Deep Edge Intelligence

Caitong Bai1, Xiaolong Cui2,* and Ai Li1
1Graduate Group, Engineering University of PAP, Xi’an, China
2Lu Mu Campus, Engineering University of PAP, Xi’an, China

*Corresponding author: bct edu@163.com

Abstract. In order to further explore and solve the problem of distributed training edge intelligent application in edge devices close to the data side, a joint training method of multi-source federated learning is proposed for the robust speech recognition model. The joint training is performed directly on the edge computing device close to the data supply side. Under the open source Chinese speech recognition data set, the recognition accuracy is 95%. Compared with federated learning, there is only 2% performance reduction, but the communication load is reduced by 90%.

Keywords: Federal learning; Robust speech recognition; Deep edge intelligence.

1. Introduction

In recent years, neural networks have been successfully applied in the field of speech recognition due to their high generalization. Among them, Chan[1] proposed LAS and applied SEquence-to-sequence to speech recognition tasks for the first time. Graves proposed CTC structure to realize real-time speech recognition. Then, Graves[2] proposed RNN-T, which introduced the attention mechanism into the model to make full use of context information. Then Jaitly[3] proposed NeuralTransducer; Then MoChA[4] was proposed. However, in the application of speech recognition model, noise interference greatly affects the effect of speech recognition, so robust speech recognition technology has developed rapidly in recent years. Pascual[5] proposed SE-GAN to introduce a judge for speech recovery against generative networks. Since the speech enhancement performance of the MMSE method improves with the accuracy of the prior signal to noise ratio (SNR) estimator used, Nicolson[6] proposed to use the remaining Long and short-term memory (RES-LSTM) network to accurately estimate the prior SNR. The MMSE method using Reslstm used subjective and objective measures of speech quality and intelligibility to evaluate the prior SNR estimator, known as DeepXi. Braithwaite[7] proposed the use of variable restricted autoencoders (VCAE) in the time domain for speech enhancement. Defossez[8] proposed that DEMUCS used convolutional neural network to build an encoder-decoder structure in the time domain to directly recover the original pure signal. Ravanelli[9] proposed the feature enhancement algorithm PASE+ based on self-supervision. However, in the context of cloud intelligence application, with the rapid growth of robust speech recognition model parameters, the annotated training data required by the training model is also increasing, and the training and reasoning calculation power required by the training model is also increasing. It presents a situation that requires high computing power and large data samples. The whole robust speech recognition
model can be divided into three parts: the denoising front end, the acoustic model and the language model. But the computing power required for each part is not suitable for centralized deployment in mobile edge devices.

With the development of cloud intelligence, its disadvantages gradually emerge: the existing intelligent applications of speech recognition are computation-based and have very strict requirements on CPU, GPU, memory, network and other resources. It is impossible to provide services for end users anytime and anywhere. Although current terminals are becoming more powerful, they are still not strong enough to support some deep learning models. For example, most voice assistants, like Apple's Siri and Microsoft’s Google’s Cortana, are cloud-based, and they won’t work properly if the network isn’t available. In addition, existing speech recognition intelligent applications generally use centralized data management, which requires users to upload their data to a central cloud-based data center. With the popularity and use of mobile devices, user data must be uploaded every time an edge device invokes AI model capabilities. However, billions of mobile users and IoT devices distributed on the edge of the network have generated and collected huge amounts of data. According to Cisco’s forecast, the data generated by mobile users and IoT devices will reach 850ZB by 2021[10]. Uploading such a large amount of data to the cloud consumes a large amount of bandwidth resources, which also creates unacceptable delays for users. On the other hand, users are increasingly concerned about their privacy. The European Union has issued the General Data Protection Regulation (GDPR)[11] to protect users’ personal information. If mobile users upload their personal data to the cloud for use in a specific intelligent application, they will face the risk of privacy leakage, that is, personal data may be extracted by malicious hackers or companies for illegal purposes. In order to solve the above problems, how to train artificial intelligence models on edge devices in the edge computing environment has become an important research direction, and the edge intelligence technology[12–14] developed from it, especially the deep edge intelligence technology for deep neural network model, has become a hot spot in recent technology development.

To solve the near the edge of the supply side edge joint training purpose of intelligent model, especially the robustness of speech recognition model, the learning techniques have been proposed, including[15] Fedavg algorithm achieved good results, but the learning technology, with large communication overhead shortcomings, in order to solve this problem, the distillation algorithms have been proposed. However, most of the training data sets of robust speech recognition models are distributed and dispersed. How to combine the distributed data and computing power to train an edge intelligent model with good performance has become an important problem. To solve this problem, a multi-source federal distillation technology is proposed in this paper. The main innovations of this paper are as follows:

1. In order to solve the problem of distributed training edge intelligence application in the edge devices near the data side, as a joint training algorithm, the distributed data and the data and computing power in the model are used to realize joint learning.

2. The model is decomposed at the edge, and then each part is distilled for federated learning, so as to avoid the problem of too much communication overhead.

3. The first attempt is to split the robust neural network first and then divide it into parts for distributed training and conduct performance comparison experiment and analysis.

2. System Design

In Figure 1, the robust speech recognition model is divided into two parts: encoder and decoder. Let $x \in \mathbb{R}^{n \times l}$ denotes the clean voice signal data set and $y \in \mathbb{R}^n$ is the corresponding label, where $n$ denotes the number of samples and $l$ is the length of each data. The training data set $D = \{(x_i, y_i)\}^m_{i=1}$ is given, in which there are $n$ pairs of speech signals and statements. Because the real environment is often full of noise, and the noise mixing situation is also complex, we let $\text{Dist}(x)$ to describe the process, and design a noise reduction front end $\mu_\alpha$ to extract features from noisy speech data, which
are parameterized by $\alpha$. Therefore, the whole process of noise reduction can also be described as an optimization problem of mapping a speech signal to a statement and defined by

$$
\min_{\alpha, \mu} \frac{1}{2} \sum_{i=1}^{n} | f_{\theta}(\mu(x)) - y_i |^2
$$

![Figure 1. The Structure of the frame](image)

Among them, the federated learning after distillation algorithm is adopted for the training of $f_{\theta}$ function and $\mu_\alpha$. By taking advantage of the good network condition of the download link, the model parameters of the download link are downloaded while logits with small number of parameters is uploaded to the upload link. The algorithm flow is shown in [16]. Then, the encoder adopts the PASE+ model in while the language model adopts Baidu language model. The target model is the decoder, and the knowledge of the language model is distilled into the decoder as a teacher network, where, the language model is trained through federal learning after distillation. The loss function of the decoder model can be expressed as:

$$
L_{CE}(\theta) = - \sum_{k=1}^{K} \delta(k, y_t) \log P_{S}(k \mid y_{t-1}, c_{t-1}, x; \theta)
$$

The output of the language model can be expressed as

$$
P_t(k \mid y_{t-1}, h_{t-1}) = \frac{\exp(z_k / T)}{\sum_{i=1}^{K} \exp(z_i / T)}
$$

Let $P_S^d = P_S(k \mid y_{t-1}, c_{t-1}, x; \theta)$, $P_t^d = P_t(k \mid y_{t-1}, h_{t-1})$, the loss function of response feedback between the language model and the decoder network can be expressed as:
\[
D_{KL}(P_T \| P_S) = - \sum_{k=1}^{K} P_T^k \log \frac{P_S^k}{P_T^k}
\]

Where, since the language model will change during the training process, the above loss function can be expressed as the Cross Entropy form between the language model and the decoder network:

\[
L_{T,S}(\theta) = - \sum_{i=1}^{K} P_T^i \log P_S^i
\]

Let \( \lambda \in [0,1] \), the loss function can be expressed as:

\[
L(\theta) = \lambda L_{T,S}(\theta) + (1 - \lambda) L_{T,T}(\theta)
\]

Combined with the above equation, it can be finally arranged as:

\[
L(\theta) = - \sum_{i=1}^{K} (\lambda \delta(k, y_i) + (1 - \lambda) P_T^k) \log P_S^k
\]

3. Experiments

3.1. Setup
The supervised data set was THCH30, Aishell-1 and ST-CMDS. Unsupervised training of language models using internal data sets. Pytorch was used as the machine learning framework of the simulation platform, Docker and Kubenetes were used for container encapsulation, Nvidia driver for Docker was used for R container calling GPU resources, and Nvidia Geforce GTX 3080 was used as GPU resources. The encoder network adopts the PASE+ model in [9], and the language model adopts Baidu pre-training model.

3.2. Discussion
As shown in the Table 1 and Figure 2, if only the encoder or decoder is federated learning, the performance will decline because more knowledge is not received. It is proved that in the edge distributed environment, the performance of integrating multiple teacher models is better than that of integrating acoustic models only. Using federated learning instead of distilled federated learning can lead to excessive traffic. The results show that the distilled federated learning is better than the federated learning.

Table 1. Algorithm comparison table

| Training Method       | Communication Cost (GB) | Accuracy |
|-----------------------|-------------------------|----------|
| FLD[17] only for Encoder | 6                       | 0.85     |
| FLD[17] only for Decoder | 5                       | 0.84     |
| FedAvg                | 100                     | 0.97     |
| Our Method            | 10                      | 0.95     |
4. Conclusions
In this paper, a robust speech recognition model training algorithm based on multi-source federated distillation is proposed. The robust speech recognition model firstly is divided into an encoder and a decoder, and then training by federated learning after distillation in order to take advantage of the data and computing power of the distributed edge device and considering the communication load. Experimental comparison on three open source Chinese speech recognition data sets: THCH30, Aishell-1 and ST-CMDS shows that, compared with traditional federated learning, the communication load is reduced by 90% while the WER is maintained at 95%.

Acknowledgments
This work was supported in part by the National Natural Science Foundation of China under Grant No. U1603261 and by the Natural Science Foundation Program of Xinjiang Province, China, under Program No. 2016D01A080.

References
[1] Chan W, Jaitly N, Le Q V and Vinyals O 2015 Listen, Attend and Spell
[2] Graves A, Fernández S, Gomez F and Schmidhuber J 2006 Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks Proceedings of the 23rd international conference on Machine learning ICML ’06 (New York, NY, USA: Association for Computing Machinery) pp 369–76
[3] Jaitly N, Sussillo D, Le Q V, Vinyals O, Sutskever I and Bengio S 2016 A Neural Transducer arXiv:1511.04868 [cs]
[4] Chiu C-C and Raffel C 2018 MONOTONIC CHUNKWISE ATTENTION 16
[5] Pascual S, Bonafonte A and Serrà J 2017 SEGAN: Speech Enhancement Generative Adversarial Network Interspeech 2017 Interspeech 2017 (ISCA) pp 3642–6
[6] Nicolson A and Paliwal K K 2019 Deep learning for minimum mean-square error approaches to speech enhancement Speech Communication 111 44–55
[7] Braithwaite D T and Kleijn W 2019 Speech Enhancement with Variance Constrained Autoencoders INTERSPEECH
[8] Défossez A, Synnaeve G and Adi Y 2020 Real Time Speech Enhancement in the Waveform Domain Interspeech 2020 Interspeech 2020 (ISCA) pp 3291–5
[9] Ravanelli M, Zhong J, Pascual S, Swietojanski P, Monteiro J, Trmal J and Bengio Y 2020 Multi-Task Self-Supervised Learning for Robust Speech Recognition ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp 6989–93
[10] Anon 2016 Cisco visual networking index: Global mobile data traffic forecast update (2017–
2022)

[11] Voigt P and Von dem Bussche A 2017 The eu general data protection regulation (gdpr) A Practical Guide, 1st Ed., Cham: Springer International Publishing 10 3152676

[12] Li E, Zhou Z and Chen X 2018 Edge intelligence: On-demand deep learning model co-inference with device-edge synergy Proceedings of the 2018 Workshop on Mobile Edge Communications pp 31–6

[13] Wang X, Han Y, Wang C, Zhao Q, Chen X and Chen M 2019 In-Edge AI: Intelligentizing Mobile Edge Computing, Caching and Communication by Federated Learning IEEE Network 33 156–65

[14] Wang Z, Cui Y and Lai Z 2019 A first look at mobile intelligence: architecture, experimentation and challenges IEEE Network 33 120–5

[15] Jeong E, Oh S, Kim H, Park J, Bennis M and Kim S-L 2018 Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data arXiv:1811.11479 [cs, stat]

[16] Park J, Wang S, Elgabli A, Oh S, Jeong E, Cha H, Kim H, Kim S-L and Bennis M 2019 Distilling On-Device Intelligence at the Network Edge arXiv:1908.05895 [cs, eess, math]

[17] Oh S, Park J, Jeong E, Kim H, Bennis M and Kim S-L 2020 Mix2FLD: Downlink Federated Learning After Uplink Federated Distillation With Two-Way Mixup IEEE Commun. Lett. 24 2211–5