September 2023

Improved Federated Learning for Handling Long-tail Words

Yuxin Ding

Yonghui Xiao

Rajiv Mathews

Mingqing Chen

Lillian Zhou

Improved Federated Learning for Handling Long-tail Words

ABSTRACT

Automatic speech recognition (ASR) machine learning models are deployed on client devices that include speech interfaces. ASR models can benefit from continuous learning and adaptation to large-scale changes, e.g., as new words are added to the vocabulary. While federated learning can be utilized to enable continuous learning for ASR models in a privacy preserving manner, the trained model can perform poorly on rarely occurring, long-tail words if the distribution of data used to train the model is skewed and does not adequately represent long-tail words. This disclosure describes federated learning techniques to improve ASR model quality when interpreting long-tail words given an imbalanced data distribution. Two different approaches - probabilistic sampling and client loss weighting - are described herein. In probabilistic sampling, the federated clients that include fewer long-tail words are less likely to be selected during training. In client loss weighting, incorrect predictions on long-tail words are more heavily penalized than for other words.

KEYWORDS

- Federated learning
- Probabilistic sampling
- Client loss weighting
- Class imbalance
- Data distribution
- Long-tail word
- Rare class
- Speech recognition

BACKGROUND

Automatic speech recognition (ASR) machine learning models are used to recognize words or phrases uttered by a user, e.g., spoken commands or queries. Such models are deployed on client devices such as smartphones, smart speakers, smart displays, or other devices that include speech interfaces, e.g., via a virtual assistant. ASR models can benefit from continuous learning and adaptation to large-scale changes, e.g., as new words are added to the vocabulary.

Federated learning can be utilized to enable continuous learning for ASR models in a privacy preserving manner. Federated learning is a machine learning technique to train a neural network on edge devices (client devices) in a decentralized manner. Federated learning enables models to be trained with strict privacy and security controls. With user permission, the model is separately trained on multiple clients and only the client gradient updates are sent to a server for aggregation. The server computes a weighted average of the client updates. The averaged gradients can be viewed as pseudo-gradients which can be applied to update the model.

An important benefit of continuous learning for an ASR model is that it ensures that the model is continuously updated to reliably detect fresh words. To update the ASR model to reflect a target distribution of words that accounts for fresh words, federated learning can be used to train the model with examples that follow this distribution. However, the distribution is typically imbalanced, skewed towards high frequency headwords. This can result in the trained model performing poorly on rarely occurring, long-tail words.

In neural network model training, one way to deal with imbalanced data is to sample the data to compensate for the imbalance. This is achieved by downsampling of over-represented data or upsampling of less represented data to create a more uniform distribution. Another common technique is to design the model loss function to penalize wrong predictions of the rare classes more heavily. This can result in the model naturally generalizing in favor of the rare class. However, these methods are usually implemented in centralized learning, and are not directly usable for federated learning.

DESCRIPTION

This disclosure describes federated learning techniques to improve ASR model quality when interpreting long-tail words given an imbalanced data distribution. Two different approaches - probabilistic sampling and client loss weighting - are described herein.

Probabilistic sampling

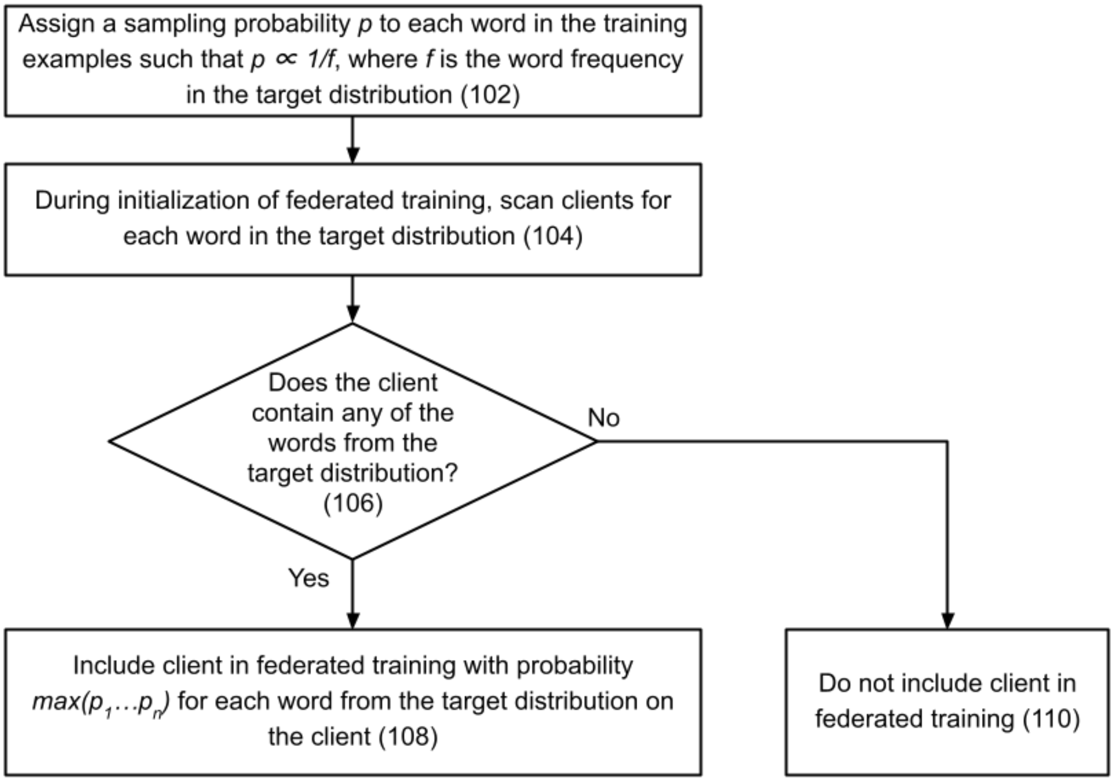
Assign a sampling probability *p* to each word in the training examples such that *p* ∝ *1/f*, where *f* is the word frequency in the target distribution (102)
During initialization of federated training, scan clients for each word in the target distribution (104)
Does the client contain any of the words from the target distribution? (106)
No
Yes
Include client in federated training with probability $max(p_1 \ldots p_n)$ for each word from the target distribution on the client (108)
Do not include client in federated training (110)

Fig. 1: Probabilistic sampling for federated training

Fig. 1 illustrates an example method for probabilistic sampling during federated training, per techniques of this disclosure. The probabilistic sampling aims to adjust the data distribution into the targeted distribution by downsampling the federated clients with only headwords.

Depending on the data distribution and the target distribution, each word in the training examples is assigned (102) a sampling probability p - $p \in (0,1)$ - such that $p \propto 1/f$, where f is the frequency of each word in the target distribution. Words that are more frequent in the distribution are assigned a smaller value of p, while words that are less frequent are assigned a higher probability.

At the start of a federated training round, client devices scan for the targeted words (104). If the client data does not contain any of the targeted words (106), the client is excluded from federated learning (110). If a client includes one or more of the targeted words (106), it is included in the federated training round with probability p_w - the probability assigned to the word w from the target word list found on the client. If the client has multiple wordlist words, then it is included in federated training with the maximum probability $max(p_{word1}, \ldots, p_{wordn})$, among the matched words from the target wordlist (108).

Client loss weighting

Fig. 2 illustrates an example method for client loss weighting during federated training. The method penalizes wrong predictions for the long-tail words more heavily.

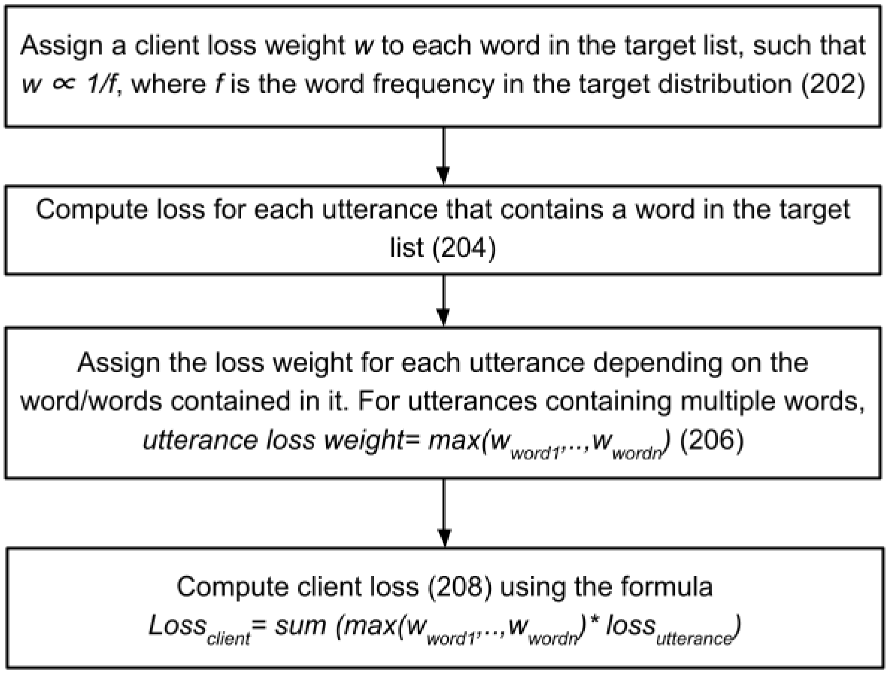
Assign a client loss weight w to each word in the target list, such that w ∝ 1/f, where f is the word frequency in the target distribution (202)
Compute loss for each utterance that contains a word in the target list (204)
Assign the loss weight for each utterance depending on the word/words contained in it. For utterances containing multiple words, utterance loss weight= max(w_word1,..,w_wordn) (206)
Compute client loss (208) using the formula Loss_client= sum (max(w_word1,..,w_wordn)* loss_utterance)

Fig. 2: Client loss weighting for federated training

Each target word is assigned a client loss weight w. w is inversely proportional to the frequency of the word; thus, more frequent words are associated with smaller client loss weights and long-tailed words are associated with larger client loss weight (202). At the client, the loss is computed for each utterance containing a word of the words in the target list (204). The utterance is assigned a loss weight, depending on the target word or words that it contains. If an utterance contains multiple target words, the utterance loss weight is the maximum among all the weights (206). The client loss is computed (208) as the sum of each utterance loss multiplied by the utterance loss weight, as indicated by the following formula:

$$Loss_{client} = \ sum\ (max(w_{word1},\ldots.,w_{wordn}\ \ ) *\ loss_{utterance})$$

The described techniques can be used to train any automatic speech recognition model to improve recognition accuracy for long-tail words. The techniques can be used to train any machine learning model with a long-tailed imbalanced data distribution via federated learning.

CONCLUSION

This disclosure describes federated learning techniques to improve ASR model quality when interpreting long-tail words given an imbalanced data distribution. Two different approaches - probabilistic sampling and client loss weighting - are described herein. In probabilistic sampling, the federated clients that include fewer long-tail words are less likely to be selected during training. In client loss weighting, incorrect predictions on long-tail words are more heavily penalized than for other words.

REFERENCES

1. McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. "Communication-efficient learning of deep networks from decentralized data." In *Artificial Intelligence and Statistics*, pp. 1273-1282. PMLR, 2017.
2. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794. 2016.
3. Shuai, Xian, Yulin Shen, Siyang Jiang, Zhihe Zhao, Zhenyu Yan, and Guoliang Xing. "BalanceFL: Addressing class imbalance in long-tail federated learning." In *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 271-284. IEEE, 2022.
4. Narayanan, Arun, Tara Sainath, Chung-Cheng Chiu, Ruoming Pang, Rohit Prabhavalkar, Y. U. Jiahui, Ehsan Variani, and Trevor Strohman. "Cascaded Encoders for Simplified Streaming and Non-Streaming ASR." U.S. Patent Application 17/237,021, filed April 21, 2021.

5. Zhao, Ding, Bo Li, Ruoming Pang, Tara N. Sainath, David Rybach, Deepti Bhatia, and Zelin Wu. "Using context information with end-to-end models for speech recognition." U.S. Patent 11,545,142, issued January 3, 2023.