Application of Federated Learning on Speech Recognition

Hao Xia and Chi Wang
Shanghai Jiao Tong University

1. INTRODUCTION

Federated learning is a new kind of machine learning method whose goal is to train a high-quality centralized model with training data distributed over a large number of clients each with unreliable and relatively slow network connections. Google proposed this new machine learning method to make user experience more acceptable and comfortable. It enables multiple mobile devices to share a prediction model or any neural network and to make changes to the original model by cooperating and collaborating. Meanwhile, all the training data will be preserved in the terminal devices and the cloud only receives an encrypted update from mobile devices. This reduces the time and energy to submit all the files and data to the server and improves users' security.

Speech recognition is one of the important applications in mobile phones and can be improved by federated learning method. Users can have personal customization while sharing the same model. The goal of our project is to see how federated learning method contributes to current speech recognition algorithm. We performed a simulative experiments on personal computers with a small data set to see the process and changes that federated learning bring.

2. BACKGROUND

Currently, speech recognition on mobile devices in one application uses exactly the same prediction model for every user. The individual characteristics have not been taken into consideration. Moreover, if the mobile device is not well-connected to the network, it is hard to realize any speech recognition. Federated learning provides a way to deal with the dilemma. The core of it is to make personalized modifications according to every user's preferences while using the same model. The key is to find a general update for the shared speech recognition model. Federated learning includes the following steps:

(1) A subset of existing clients is selected, each of which downloads the current model;
(2) Each client in the subset computes an updated model based on their local data;
(3) The model updates are sent from the selected clients to the sever;
(4) The server aggregates these models (typically by averaging) to construct an improved global model.

In particular, there are many users logging in the system and making the updates all at once and they do it randomly. So parallelism is another important feature in the federated learning problem. Federated learning balances the simultaneous feedback and optimizes the combination of their updates.

Speech recognition is an interdisciplinary subject. Nowadays, the speech recognition algorithms have become more and more mature. It mostly consists of two parts: the acoustic model and the language model. The former part receives a short vocal message and returns syllables or corresponding phonetic symbols while the latter part transfers the pronunciation in to actual literal words by checking the probability dictionary. The acoustic model includes

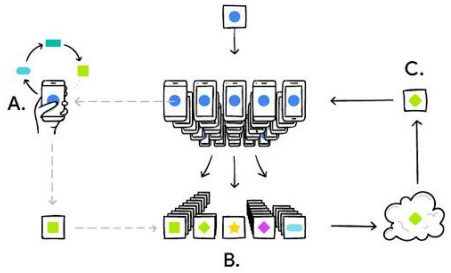

Fig. 1. Federated Learning

many detailed procedures such as the read-in of an audio file and the character extraction of it. The most commonly used method to process an audio file is to generate a statistic model based on time series like HMM. The language model calculates the probabilities of the occurrences of words and sentences. This is affected by the feature of the language itself and the habit of the speaker. In addition, noise detection has been included in most of the speech recognition models. The federated learning can help it gain more popularity by providing a more personalized service. By collaborating, the training of the speech recognition model will be more accurate and diversified.

There are a lot of speech recognition algorithm in applications or in researches. By implementing speech recognition with federated learning, users will acquire an off-line speech input and test with improved accuracy. The updates will be submitted once they are online again. From the users' perspective, the process can be viewed as below:

(1) Download (or use the embedded) current speech recognition model on mobile phone;
(2) Train the model with local data by using speech input and correcting the prediction made by the model in daily life. During the process, the model gradually changes to cater to the user's expectations. The model first adjusts to the pronunciation of the speaker and then generates a mildly-changed prediction language model;
(3) The mobile phone generalizes a small specialized update for the model according to the change of the model;
(4) Send the update back to cloud with encryption;
(5) The shared model is modified according to the update;
(6) The new model is downloaded.

One thing to notice is that the initialization of models on mobile devices would affect the aggregation of the models aggressively. In related work Communication-Efficient Learning of Deep Networks from Decentralized Data, it showed that using the same weight to combine the models, using the same initialization would generate a obviously-seen decline in loss. This means that after each iteration, the models on the mobile devices had better be updated as the same. In light of this, we sent the updated model to each user as well in our project.

3. MOTIVATION

Federated learning is a new concept and is expected to have many applications as mobile phones are important to people these days. Oral language is more personalized than written message or text-entry and requires the speech recognition model to be more specific to the users. Federated learning can make this happen by modifying models on mobile devices.

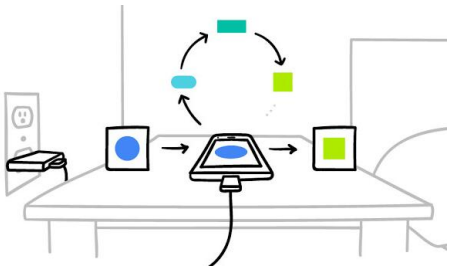

Fig. 2. Local Training

There are problems to be solved in this method. To start with, one of the most important procedure in federated learning is to modify the shared model in an appropriate way. This means that the algorithm should be able to aggregate all the updates without making too radical changes to the original model (because the shared model should be a generalized network instead of a personalized one) and wasting too much resources and time (because there may be a lot of mobile devices submitting their updates at the same time).

Secondly, the communication between the mobile phones and the cloud or the server should have low latency and fewer updates times. This means that the algorithm should judge whether the updates are obvious enough to make a change to the original model and are not too big to slow down the transferring process.

Also, the training process is done on the mobile devices instead of a cloud server in federated learning. As a result,running time and power consumption should be low because a mobile phone have weaker computational ability. The data is from the user, too. This means that the training is not contiguous and only proceeds when the users are trying the speech recognition application.

4. PROBLEM FORMULATION

In this section, we describe the federated problem formally. We view a neural network model as a matrix who consists of all the cells and parameters of the cells in it. According to the steps of federated learning above, all the mobile device users(clients) have a copy of the neural network, which is the same as the current matrix. And the problem goes as follows:

(1) Assume the original model with parameters embodied in a real matrix W;

(2) The server distributes the current model W to a subset S of n clients. These clients independently train the model based on their local data;

(3) Let the new local models be W_i, so the update of client i can be written as $H_i = W_i - W$, for $i \in S$;

(4) Each selected client then sends the update back to the sever where the global update is computed by aggregating all the client-side updates: $W_{updated} = W + \eta H$, H is a combination of all H_i.

In particular, the matrix W is adjusted to the model and it can represent the parameters of each layer in the neural network. To represent the whole network is to combine the matrices together which is multiplication in our case since the network is sequential. But the multiplication is trivial because it doesn't have to be done. The change only has to be made to the parameters of each layer.

To implant this on speech recognition, the limit to the transferring of the updates can not be overlooked since all the neural network of speech recognition is relatively huge. The first idea to overcome this is showed in the formulation above. To improve efficiency, previous paper proposed structured update, which ensures low rank to H, and sketched update, which compresses each individual's update by sub-sampling. We adopted the idea of both methods, i.e. making the matrices more light to ensure efficiency of transferring, but we used a more crude way due to the limit of the computational ability of our computers. We applied greedy algorithm to this problem and set the change to zero(no change) if its corresponding absolute value is smaller than a threshold. This makes the matrices more sparse and sparsity, which can be manipulated by more complex data transfer method, is a very important feature to high transfer rate.

Another innovative idea of our algorithm is to make the combination of all the update matrices related to the number or scale of changes they conclude. For example, if some of the results of the speech recognition function appear more frequently than others, then their corresponding updates will be taken more into consideration because this means the changes is generated due to some popular language habits. On the other hand, the most acute change will be ignored or diminished when combining with other update matrices because it is most likely to be a very personal change.

5. PROPOSED MODEL

We found a speech recognition model using time-series neural network online and its parameters is easy to obtain. The function of the model includes training models from given audio files, testing results by models and saving as well as loading models which are both useful in federated learning. The speech recognition algorithm also includes the read-in and feature extraction (the feature is MCFF feature). In our project, in order to simply the problem and stress the point of the influence of the federated learning, we focuses on the modification of the acoustic part of the speech recognition model while the language model is trivial.

In order to simulate the situation where the federated learning is used, we set up a server to be the cloud. The different clients or mobile phone users are mimicked by various processes. The whole procedure goes like this:

(1) The server is constantly running the whole data set and testing the global loss function to check the model's accuracy while monitoring the number of the users(processes);

(2) When the number reaches five, the users begin to receive missions to train the speech recognition model on their own while others join them. The data set for each process is chosen randomly, which is like the actual situation.

(3) The server collects the updates from them and test it by running the whole data set. The loss is calculated for this.

As for the federated learning part, the model follows as the problem formulation in the previous section and the above procedure. One interesting point to notice is that the model focuses on the communication between the server and the processes. The server and the clients are constantly in communication in our model. This can simulate the fact that whenever a mobile device is back online,

a connection is set up between them. However, it is not until the updates number reaches the designed value that a general change is made for the public model. This can reduce the rounds of the communication. The new model would be again downloaded to every clients after the change is updated.

6. EXPERIMENTS

We used an open-source data set thchs30. It is a Mandarin speech data set which includes the audio files in the form of wav, the phonetic syllables with the right pronunciation and the correct text corresponding to it. As mentioned before, we focused on the audio part, i.e. we only check whether the pronunciation is correct and the following graphs(Figure.3-Figure.6) are based on that.

When the updates are received and completed, the server will run the whole data set by testing the model and calculate the total loss. The rounds means the number of times that the clients make a collaborative update. Notice that all clients are in the process of giving updates in our experiments. In a larger system with more clients, few off-line users who fail to update their model are ignored by us.

Fig. 3. Loss Through Round

Fig. 4. Loss Through Time

According to the graph above, the loss of the whole model is on the decline and tend to be convergent. The wave in the end may be because that the number of the layers of the speech recognition model are not very large and the influence of one update is still obvious.

The following two graph shows that the accuracy is gradually improved. This means by training some of the data in various and multiple clients can collaboratively update the original model into a more general one if the model is trained from scratch. See the curve at the end, the accuracy has gone down a little which is a evidence that the updates can affect the original model but maybe a little too large.

As for the time consumption, most of the time is used on the calculation on mobile devices and the time used on the updates transferring is relatively small, which satisfies the expectations.

Fig. 5. Accuracy Through Round

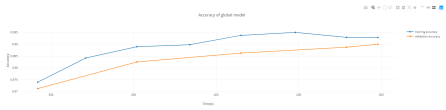

Fig. 6. Accuracy Through Time

7. CONCLUSION

Federative learning is a new type of machine learning that emphasizes on the users' perspective. It gives the users a more personalized experience while give the server a chance to synthesis all the feedback from the clients.

In this project, we implemented federative learning and tried it on speech recognition. We provide a perspective for the speech input on mobile phones to be realized collaboratively. By experiments, we found that training a speech recognition model by different clients randomly can help the general training of the whole model. The algorithm realized the less transferring time. However, it also raises realistic problems like the computational incompetence of the mobile devices. There is still work to be done if people want to apply federated learning to mobile phones.

There are also some improvements to be made. We did not check the accuracy for each process from the server because the data set for each of them is chosen randomly. This contradicts the situation where a typical voice and oral habit is used for a single user. We may solve this problem by finding specific language data set. Another perspective to improve the federated learning is to preserve most of the local change. Though it is the best to initialize as the same but it can't preserve the users' preferences after each iteration. So instead of updating the whole model for the mobile devices, a new averaging method can be found to make the change more appropriate.

REFERENCES

J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

D. Rehak, P. Dodds, and L. Lannom, "A model and infrastructure for federated learning content repositories," in *Interoperability of Web-Based Educational Systems Workshop*, vol. 143. Citeseer, 2005.

[1] [2]