Federated Learning for Emotion Recognition

Ivan Chau (ic2504) and Daniel Garces (dg3008)

Executive Summary

Goal: implement a federated learning system for emotion recognition in speech that will adapt to a specific edge devices' user while also contributing to the construction of a shared global model that will provide a more accurate starting point for new users (as it contains the combined information of all the users utilizing the system).

Solution approach: Use Tensorflow implementation of federation to create a custom federated learning system that process speech in different clients and then aggregates the models into a combined model being run in a centralized server.

Value of the proposed solution: by combining fine-tuned models from different users we can generate a model with better predictive accuracy that is more robust towards intonation and pitch variations across users

Motivation

- Emotion recognition in speech is a very difficult task due to variations in pitch and intonations across different groups of people
- There are very few datasets for emotion recognition, so supervised approaches tend to not generalize well to real-life data
- A system that is being constantly improved by user interaction could lead to more robust emotion recognition models that address the issue of the variation in speech patterns that make it difficult to generate a highly accurate emotion recognition model

Background

To help the reader understand the proposed solution, we will briefly cover the following concepts:

- Mel Scale
- Mel Frequency Cepstral Coefficients
- LSTM
- Federated Learning
- Previous Work

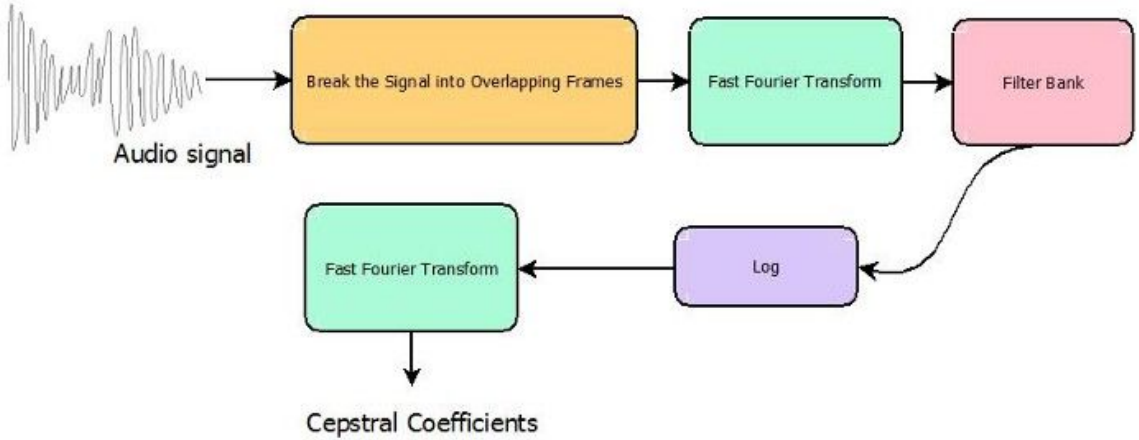
Mel Scale

Mel scale is a scale that relates the perceived frequency of a tone to the actual measured frequency. It scales the frequency in order to match more closely what the human ear can hear (humans are better at identifying small changes in speech at lower frequencies). This scale has been derived from sets of experiments on human subjects.

$$\text{Mel}(f) = 2595 \log\left(1 + \frac{f}{700}\right)$$

Mel Frequency Cepstral Coefficients

Any sound generated by humans is determined by the shape of their vocal tract (including tongue, teeth, etc). If this shape can be determined correctly, any sound produced can be accurately represented. The envelope of the time power spectrum of the speech signal is representative of the vocal tract and MFCC (which is nothing but the coefficients that make up the Mel-frequency cepstrum) accurately represents this envelope. In our project, the filter bank is composed of 13 linear filters and 27 logarithmic filters.

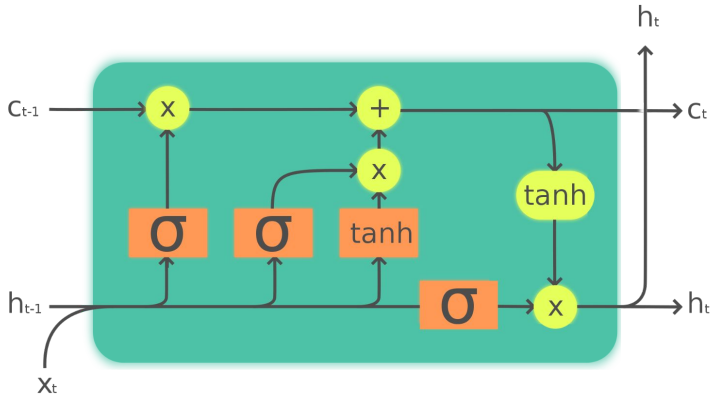


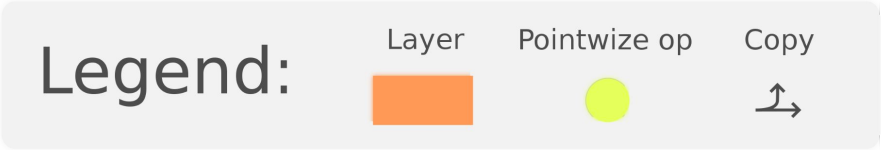
Audio Preprocessing

- The audio files are split in short-term windows (0.2 seconds) and the Mel Frequency Cepstral Coefficients (MFCCs) for each window in a signal are calculated. These features (represented as vectors of doubles) are later stacked to get the representation of the entire signal.

- 40 MFCCs are extracted for each window (corresponding to 13 Linear Filters and 27 Log Filters)

LSTM: Long-Short Term Memory





- Each cell block has an input, output, and forget gate
- Useful for learning sequence data, like audio, because information can persist over time
- Different backpropagation (with respect to time) procedure.

Previous Work - emotion recognition

- Goel et al (2020): Emotion classification using transfer learning for multiple languages using IEMOCAP as the base dataset for training.
- Tripathi et al (2019): Also uses IEMOCAP, including motion capture, speech, and text. They identify best architectures for classification.
- Zhou et al (2020): Emotion classification using transfer learning from Automatic Speech Recognition (TEDLIUM model v2), utilizing IEMOCAP as the fine-tuning dataset
- Ananthram et al (2020): Multi-modal emotion classification using transfer learning from Speaker Recognition (Vox-Celeb v2 model), utilizing IEMOCAP as the testing dataset

Our project utilizes architectures similar to the ones used in Tripathi et al for the speech module, but instead of running a centralized model, we integrate these models to the Federated Learning system proposed by Konečný et. al (2016)

Federated Learning

Move learning onto edge devices. Get updates, keep data on devices.

We can choose to train in a decentralized fashion, per client, with each client having a potentially differently distributed dataset.

- What are the impacts with a change in client dataset distribution?
- How does this compare to standard deep learning procedure, empirically?
- What types of neural architectures perform the best for this task?
- How to test all of this in practice with modern day technologies?

Technical Challenges

- Running a simulation for Federated Learning on one machine can be costly computationally for a large number of clients / data
  - Model adjustments?

- GPU Tensorflow compatibility, computational bottlenecks

- Will we converge to the same model given the FL approach?
  - What is a reasonable comparison between these two approaches?

- How to split the data effectively -- which is the best approach?

Approach Overview

- Conduct Federated Learning with Tensorflow Federated.

- Optimize audio sequence processing in the IEMOCAP dataset with Mel Frequency Cepstral Coefficients to generate a vectorized dataset.

- Compare performance with the baselines in the centralized training among these metrics:
  - Time to train
  - F1-Scores, Overall accuracy
  - Confusion Matrices

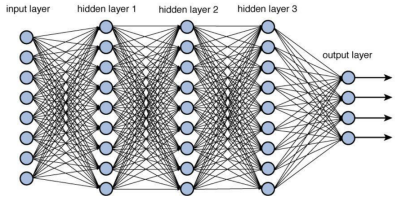
Approach Details

We will introduce more variables in our experiments:

- Impact of number of clients?
- Impact of different architectures on performance and runtime?
- Impact of different distributions of the IEMOCAP dataset for each client?

Solution Architecture

Deep Neural Network

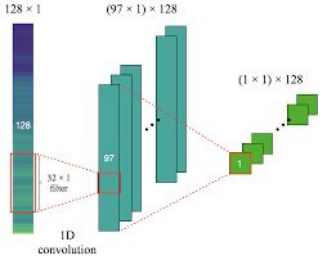


Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv1d_4 (Conv1D)	(None, 98, 512)	52736
dropout_5 (Dropout)	(None, 98, 512)	0
flatten_1 (Flatten)	(None, 50176)	0
dense_2 (Dense)	(None, 512)	25690624
dropout_6 (Dropout)	(None, 512)	0
embedding_layer (Dense)	(None, 512)	262656
dropout_7 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 4)	2052

Total params: 26,008,068
Trainable params: 26,008,068
Non-trainable params: 0

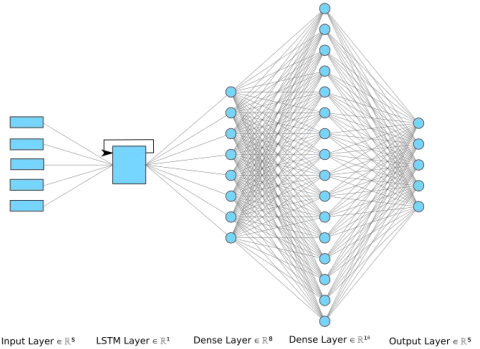
ConvNet



Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 98, 512)	52736
dropout (Dropout)	(None, 98, 512)	0
conv1d_1 (Conv1D)	(None, 96, 128)	196736
dropout_1 (Dropout)	(None, 96, 128)	0
conv1d_2 (Conv1D)	(None, 94, 128)	49280
dropout_2 (Dropout)	(None, 94, 128)	0
conv1d_3 (Conv1D)	(None, 92, 64)	24640
dropout_3 (Dropout)	(None, 92, 64)	0
flatten (Flatten)	(None, 5888)	0
embedding_layer (Dense)	(None, 256)	1507584
dropout_4 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 4)	1028

Total params: 1,832,004
Trainable params: 1,832,004
Non-trainable params: 0

LSTM



Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 100, 512)	1120256
lstm_1 (LSTM)	(None, 512)	2099200
embedding_layer (Dense)	(None, 512)	262656
dense (Dense)	(None, 4)	2052

Total params: 3,484,164
Trainable params: 3,484,164
Non-trainable params: 0

Implementation Details

Everything implemented in Python on Colab -- due to machine computing constraints, mix of CPU 20 epochs (~16GB RAM) and GPU 100 epochs (NVIDIA V100)

- Data preprocessing to get Mel Frequency Vectors
- Models implemented in tf.keras
  - Federated Averaging with Adam
    - Client lr: 1e-4
    - Server lr: 1e-2
  - Batch size of 20 per client
- Tensorflow Federated
  - Generating Federated Data per client for different data distributions
  - Creating iterative_process training loop
- Evaluation code complete with sci-kit metrics and Keras.

Experiment Design Flow

For our project we consider the following three experimental setups:

- Emotion Classifiers with a centralized model. This setup was used as a baseline for comparison with the federated setups
- Emotion Classifiers with a federated model and data randomly assigned to each client. For this setup we considered two scenarios:
    - we used an arbitrary number of clients (5) and we split the data evenly among them
    - we utilize only two of the architectures (convolutional and deep) and vary the number of clients to evaluate the effects of the number of clients on the F1 score.
- Emotion Classifiers with a federated model and a single emotion assigned to each client. With IEMOCAP we are only considering 4 emotions as the other 2 emotions have less than 25 utterances associated with them. Therefore, we used 4 clients each with data for a single emotion (anger, sadness, excitement, neutral).

Experiment Design Flow

In all of these setups, we are considering three types of architectures for the neural networks:

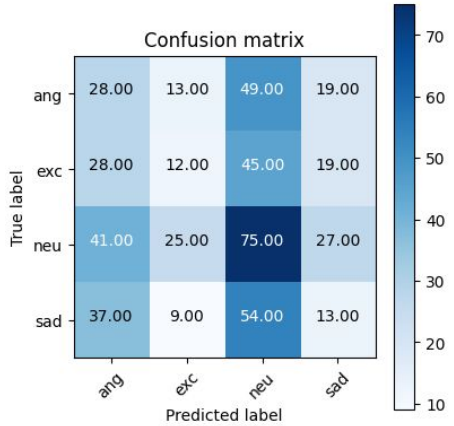
1. Deep Network
2. Convolutional Network
3. Recurrent Network

We train each architecture for each setup for 20 epochs (or 20 rounds for the federated learning system), and then record the training time, and the accuracy, precision, recall, and F1 score on the testing set. The results of these experimental setups are shown in the next slides

Results (Deep Network)

[INFO] evaluating network...

	precision	recall	f1-score	support
ang	0.21	0.26	0.23	109
exc	0.20	0.12	0.15	104
neu	0.34	0.45	0.38	168
sad	0.17	0.12	0.14	113
accuracy			0.26	494
macro avg	0.23	0.23	0.22	494
weighted avg	0.24	0.26	0.24	494

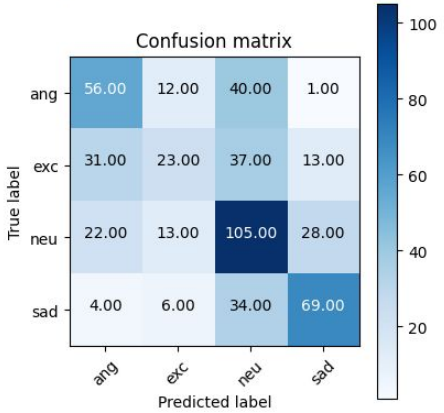


Baseline Results

[INFO] evaluating network...

	precision	recall	f1-score	support
ang	0.50	0.51	0.50	109
exc	0.43	0.22	0.29	104
neu	0.49	0.62	0.55	168
sad	0.62	0.61	0.62	113
accuracy			0.51	494
macro avg	0.51	0.49	0.49	494
weighted avg	0.51	0.51	0.50	494

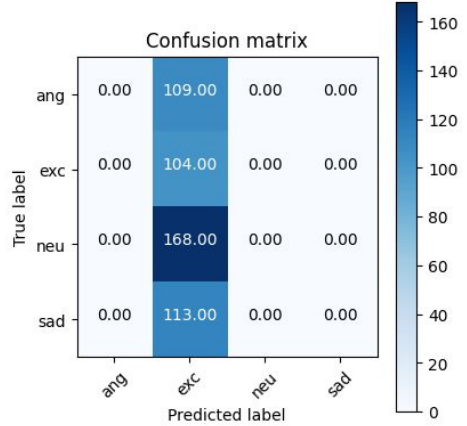
Training time: 626.9377512931824



Random Assignment per Client Results

	precision	recall	f1-score	support
ang	0.00	0.00	0.00	109
exc	0.21	1.00	0.35	104
neu	0.00	0.00	0.00	168
sad	0.00	0.00	0.00	113
accuracy			0.21	494
macro avg	0.05	0.25	0.09	494
weighted avg	0.04	0.21	0.07	494

Training time: 479.5608365535736



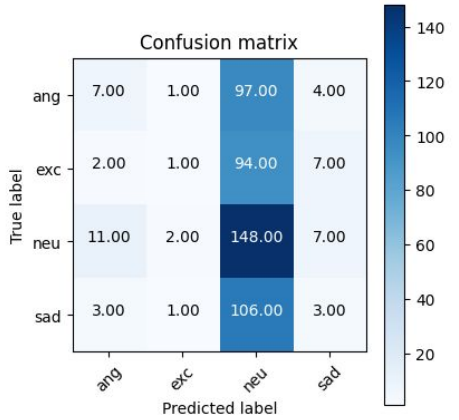
Single Emotion per Client Results

Results (Convolutional Network)

```
[INFO] evaluating network...
              precision    recall  f1-score   support

         ang       0.30      0.06      0.11       109
         exc       0.20      0.01      0.02       104
         neu       0.33      0.88      0.48       168
         sad       0.14      0.03      0.04       113

    accuracy                           0.32       494
   macro avg       0.24      0.25      0.16       494
weighted avg       0.26      0.32      0.20       494
```



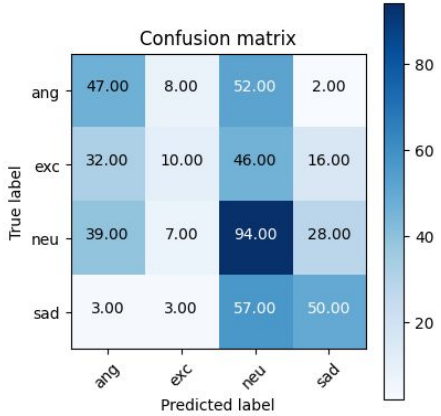
Baseline Results

```
[INFO] evaluating network...
              precision    recall  f1-score   support

         ang       0.39      0.43      0.41       109
         exc       0.36      0.10      0.15       104
         neu       0.38      0.56      0.45       168
         sad       0.52      0.44      0.48       113

    accuracy                           0.41       494
   macro avg       0.41      0.38      0.37       494
weighted avg       0.41      0.41      0.38       494

Training time: 315.8074586391449
```



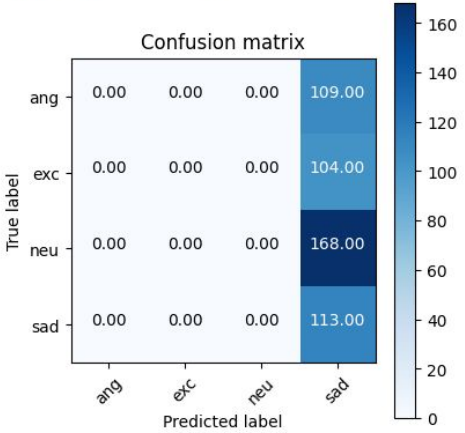
Random Assignment per Client Results

```
              precision    recall  f1-score   support

         ang       0.00      0.00      0.00       109
         exc       0.00      0.00      0.00       104
         neu       0.00      0.00      0.00       168
         sad       0.23      1.00      0.37       113

    accuracy                           0.23       494
   macro avg       0.06      0.25      0.09       494
weighted avg       0.05      0.23      0.09       494

Training time: 277.5395712852478
```



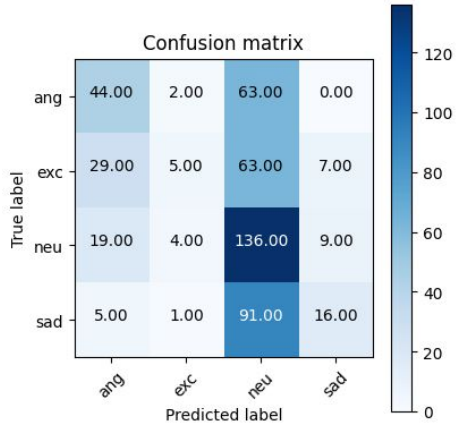
Single Emotion per Client Results

Results (Recurrent Network)

```
[INFO] evaluating network...
              precision    recall  f1-score   support

         ang       0.45      0.40      0.43       109
         exc       0.42      0.05      0.09       104
         neu       0.39      0.81      0.52       168
         sad       0.50      0.14      0.22       113

    accuracy                           0.41       494
   macro avg       0.44      0.35      0.31       494
weighted avg       0.43      0.41      0.34       494
```



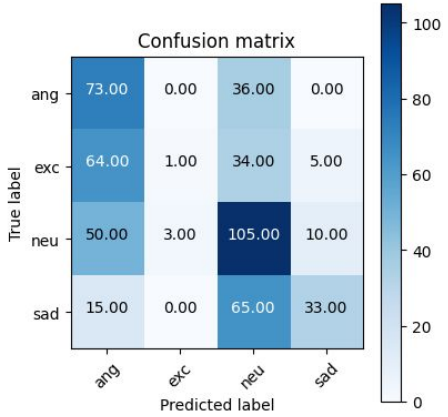
Baseline Results

```
[INFO] evaluating network...
              precision    recall  f1-score   support

         ang       0.36      0.67      0.47       109
         exc       0.25      0.01      0.02       104
         neu       0.44      0.62      0.51       168
         sad       0.69      0.29      0.41       113

    accuracy                           0.43       494
   macro avg       0.43      0.40      0.35       494
weighted avg       0.44      0.43      0.38       494

Training time: 5407.064162969589
```



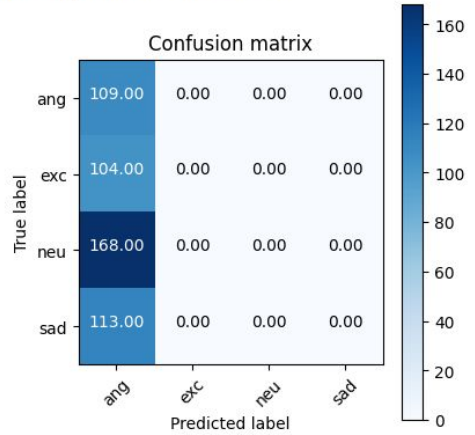
Random Assignment per Client Results

```
              precision    recall  f1-score   support

         ang       0.22      1.00      0.36       109
         exc       0.00      0.00      0.00       104
         neu       0.00      0.00      0.00       168
         sad       0.00      0.00      0.00       113

    accuracy                           0.22       494
   macro avg       0.06      0.25      0.09       494
weighted avg       0.05      0.22      0.08       494

Training time: 4645.199693202972
```



Single Emotion per Client Results

Results (Summary Deep Network)

Metric	Baseline Results	Random Assignment per Client Results	Single Emotion per Client Results
Accuracy	0.26	0.51	0.21
Precision	0.23	0.51	0.05
Recall	0.23	0.49	0.25
F1 Score	0.22	0.49	0.09
Training time (sec)	480	627	480

Results (Summary Convolutional Network)

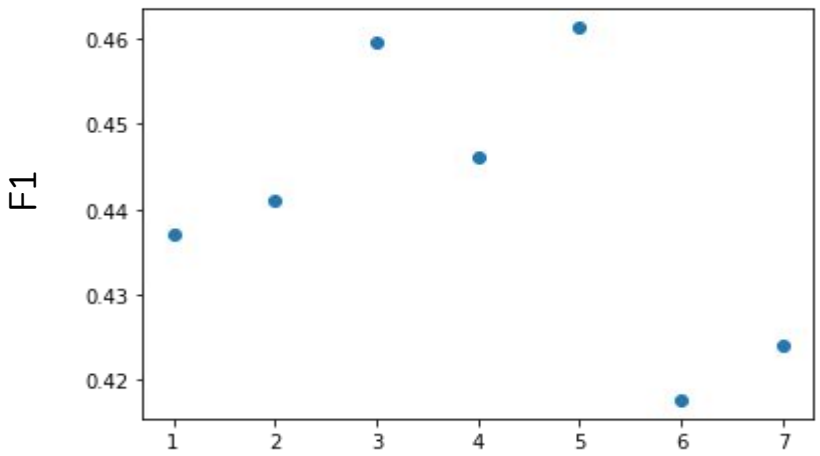
Metric	Baseline Results	Random Assignment per Client Results	Single Emotion per Client Results
Accuracy	0.32	0.41	0.23
Precision	0.24	0.41	0.06
Recall	0.25	0.38	0.25
F1 Score	0.16	0.37	0.09
Training time (sec)	460	316	277

Results (Summary Recurrent Network)

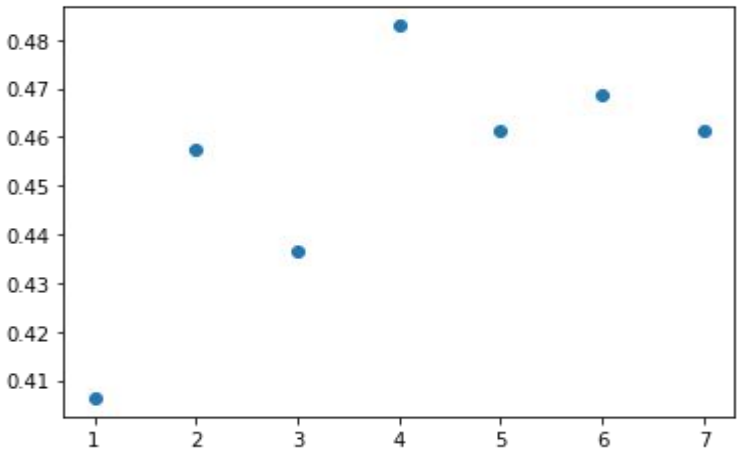
Metric	Baseline Results	Random Assignment per Client Results	Single Emotion per Client Results
Accuracy	0.41	0.43	0.22
Precision	0.44	0.43	0.06
Recall	0.35	0.40	0.25
F1 Score	0.31	0.35	0.09
Training time (sec)	5740	5407	4645

Results (Effect of # of Clients on F1 score)

DNN



CONV



- Federated Model trained for 10 rounds
- Each client has a randomized subset of the data, total size of all datasets = size of one epoch

Future Work

- More simulations where data is partitioned across different speakers
- Inclusion of a testing module where the emotion classification happens in real-time and the user has the opportunity to qualify the classification so the system could use this labels to perform supervised online learning
- Extend the work to a multimodal setting, where both speech and text are considered so that the system could be extended to other languages where intonation denotes meaning rather than emotion

Conclusion

- Federated Learning can converge to similar accuracies compared to the centralized model
- Diversifying the client dataset is key to quicker convergence
- Increasing the number of clients helps to a degree, but the benefit is reduced later
- Have these in mind when developing FL:
    - Choose your neural architecture carefully, as it may impact how long it takes to converge to certain accuracies
    - Choose your simulation methods carefully, as there are compute bottlenecks