



Training a Language Model using Federated Learning



Pierre Philbert · Follow

Published in Linagora LABS · 5 min read · Nov 16, 2020



548



1



I work as a trainee in LINAGORA's R&D department on LinTO project. If you do not know LinTO, I strongly recommend you look into it. Smart voice-based assistants are not new, and most of us knows current leaders (Alexa, Google Home, ...) but none of them, except LinTO, are open-source.

This article aims to present federated learning through a brief explanation and an application to a simple case of Natural Language Processing (NLP).

Why ?

Why has LINAGORA had a team [of engineers and data scientist] working on a voice assistant for the last two years when it's possible to use the Google offering ? What was the motivation in first place to spend this amount of energy, time, and money ? It is LINAGORA's mission to fight for digital sovereignty, and for user's privacy while improving user experience.

How ?

To achieve their goal, they are offering plenty of services, but I will only talk about the R&D team. One of their objectives is to assure that LinTO will be specific to the realm of the client. For instance, if an electricity supplier company pays for LinTO's services, LINAGORA will make sure to integrate the specific vocabulary of the electricity supplier business. It enables them to automate activities of low added value tasks (i.e. writing a meeting report) without depending on GAFAM.

What ?

Here comes the technical part: if you want to learn what the elementary components of a smart voice-based assistant are, I recommend you to read [this article](#) — from now on I will assume you are familiar with these components.

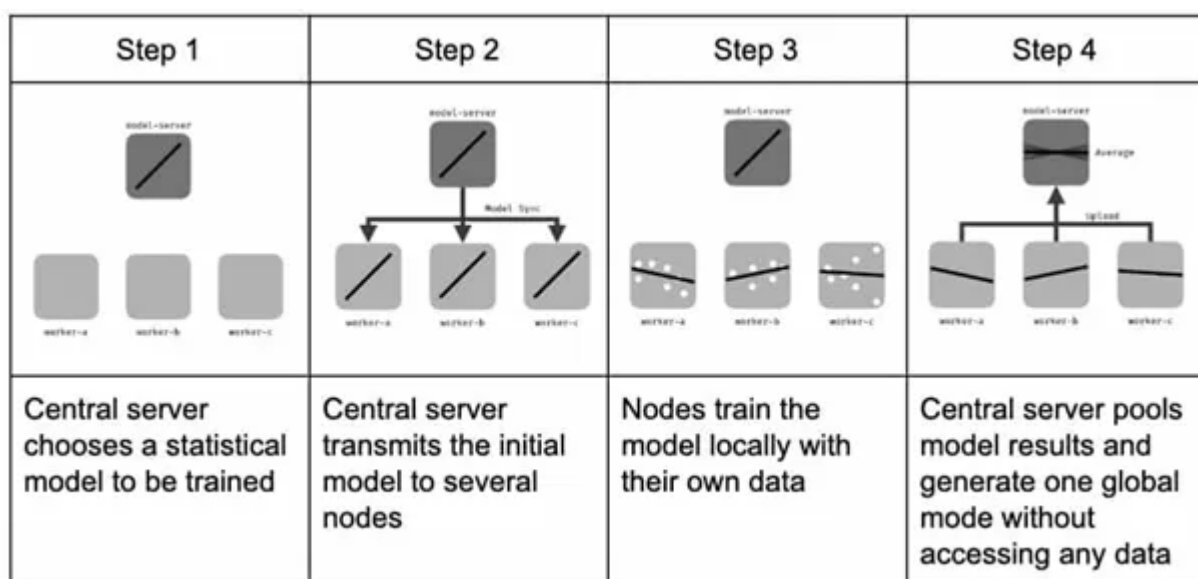
If you look to train a language model or/and speech-to-text , you might read these two articles ([LSTM for Text Generation](#) — [Speech To Text Model](#)) . You will see that to do so, you will need to work with different kind of datasets containing voice recording and their transcriptions. The data are stored on a server where you will train and test your model.

But as it is stipulated in article 4 of General Data Protection Regulation (GDPR), transcripts of discussion might be private and voice recordings are

considered, no matter what is being said, as highly sensitive data. Meaning it is very delicate to collect, store and manipulate these essential data.

The problem I focused on during my internship was the following : **how do you learn on extensive datasets without being intrusive ?**

It turns out that Google already started to work on this question in 2017 when they published their first article on federated learning. Instead of collecting all users' information at one location (on a server) and updating weights locally, the model is sent to all users and train separately on those users' device's data. Once it is done you take the weights average of your neural network and send it to the server. To sum up, you will never access your clients' personal data, neither the model of each client after training but only the average of all clients' model who take part in training.



Federated learning explanation

I will present to you an application of a distributed algorithm using Tensorflow Federated framework (also known as tff) to train a Speech-To-Text (STT) model. I followed a tutorial and used four convolutional layers to recognize ten different words : yes, no, up, down, left, right, on, off, stop, go.

I will explain what I did step by step to achieve this mission :

STEP 1 — Prepare the dataset

The first step will be to process the audio files to make them understandable to your Artificial Intelligence (AI). Once the sampling, and conversion part is done, you can split your dataset into a 80% training and 20% validation.

```
x_tr, x_val, y_tr, y_val = train_test_split(np.array(all_wave), ... )
```

Split

To simulate a federated process, you can use an array where each element will be a Map dataset representing a client. In my case, to simplify the process, I use an IID (independent and identically distributed) subset.

```
# MAKE FEDERATED DATASET / PREPARE DIFFERENT CLIENTS
def g(x):
    a = MARK_FOR_BATCH * 5000
    b = a + 5000
    return x_tr[a:b],y_tr[a:b]

def generate_data():
    global MARK_FOR_BATCH
    result = []

    # CONSTRUCT ARRAY OF MAPDATASET
    for i in range(3):
        result.append(dataset.map(g).take(15))
        MARK_FOR_BATCH += 1

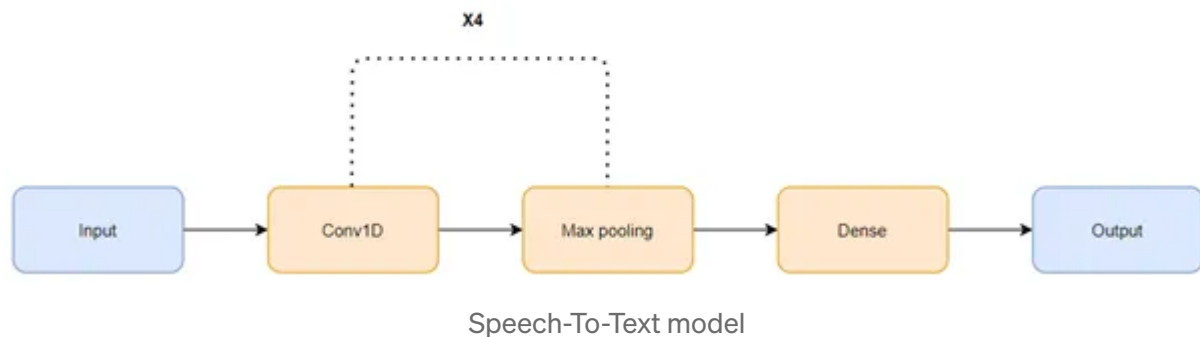
    return result

federated_data = generate_data()
#####
```

Generate Clients

STEP 2 — Create model

Using Keras you will define a model that will correspond to this schema :



STEP 3 — Create a wrapper and an iterative process

The wrapper will return a Tensorflow Federated object containing Keras model, the input specification and the loss function.

To define your iterative process, you will use your wrapper, a client optimizer to compute local model updates on each client and a server optimizer that will apply the averaged update to the global model at the server. By default, the server optimizer is a Stochastic Gradient Descent (SGD) with a learning rate of 1.

```

# CREATE FL WRAPPER
def create_tff_model():
    input_spec = federated_data[0].element_spec
    keras_model_clone = tf.keras.models.clone_model(model)
    return tff.learning.from_keras_model(
        keras_model_clone,
        input_spec=input_spec,
        loss=tf.keras.losses.CategoricalCrossentropy(from_logits=True))

# DEFINE ITERATIVE PROCESS
fed_avg = tff.learning.build_federated_averaging_process(
    model_fn=create_tff_model,
    client_optimizer_fn=lambda: tf.keras.optimizers.Adam())
  
```

Wrapper

STEP 4 — Training

The iterative process that you have defined in step 3 only have two functions
:

`initialize()` : The computation takes no arguments and returns one result — the representation of the state of the Federated Averaging process on the server.

`next()` : It represents a single round of Federated Averaging, which consists of pushing the server state (including the model parameters) to the clients, on-device training on their local data, collecting and averaging model updates, and producing a new updated model at the server.

You will use the two following functions for training:

```
# TRAINING

state = fed_avg.initialize()

for i in range(EPOCH):
    state, metrics = fed_avg.next(state, federated_data)
```

Train

Do not forget to save your model so you can evaluate it.

```
filepath = "./weights-improvement-"+str(i+1)+"-"+str(metrics.loss)+".hdf5"
tempo_model = tf.keras.models.clone_model(model)
tff.learning.assign_weights_to_keras_model(tempo_model, state.model)
tempo_model.save(filepath)
del tempo_model
```

Save Model

To evaluate your model, just compute accuracy :

```

# load model
model = keras.models.load_model('weights-improvement-5000-1.5704393.hdf5')
model.compile(loss='categorical_crossentropy',optimizer='adam')

# predict label given audio
def predict(audio):
    prob=model.predict(audio.reshape(1,8000,1))
    index=np.argmax(prob[0])
    return classes[index]

# compute accuracy of model
count = 0
size = len(x_val)
for index in range(size):
    sample = x_val[index].ravel()
    if classes[np.argmax(y_val[index])] == predict(sample):
        count += 1

print("accuracy ",count/size)
-----output-----
accuracy 0.8939713816561107

```

Evaluation

CONCLUSION

For my federated approach, I have simulated 3 clients each having 15 audio samples and trained them for 5000 epochs. According to the validation dataset, I have reached an accuracy of 0.89, overtaking the standard approach using the same model with the same amount of data. The only constraint is the training time : using 12 CPUs, I needed 36 hours to finish training when I only needed 15 minutes with standard approach. This could be explained by the needs of GPU that require TFF.

	Federated	Centralize (batch = 1)	Centralize (batch = 32)
Trainable params	1 611 498		
Audio files per epoch	45	17 049	17 049
Epochs	5000	13	13
Total audio files used for training	225 000	225 000	225 000
Training time (12 CPUs)	36 hours	44 minutes	15 minutes
Accuracy	0.89	0.17	0.85
User's privacy respected	YES	NO	NO

Comparison Table

- Linto
- Federated Learning
- Speech Recognition
- NLP
- Gdpr Compliance




Written by Pierre Philbert

6 Followers · Writer for Linagora LABS

Follow

More from Pierre Philbert and Linagora LABS



 Sonal Sannigrahi in Linagora LABS

Next Word Prediction: A Complete Guide

As part of my summer internship with Linagora's R&D team, I was tasked with...

Sep 7, 2020  313  1

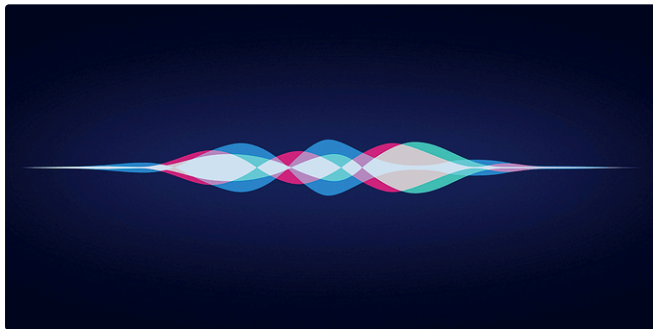


 ons wechtati in Linagora LABS

Construction of 360° images dataset for image recognition

During my summer internship with Linagora's R&D team, I was immersed in a world that an...

Mar 1, 2021  327




 Rudy BARAGLIA in Linagora LABS

Voice Activity Detection for Voice User Interface.

As a part of a R&D team at Linagora, I have been working on several Speech based...

Jun 20, 2018  1.5K  1



 Lea Baviere in Linagora LABS

Training of a speech recognition model for the Spanish language

This paper provides an overview of the development of a Speech Recognition Mode...

Sep 16, 2020  462  1




See all from Pierre Philbert

See all from Linagora LABS

Recommended from Medium

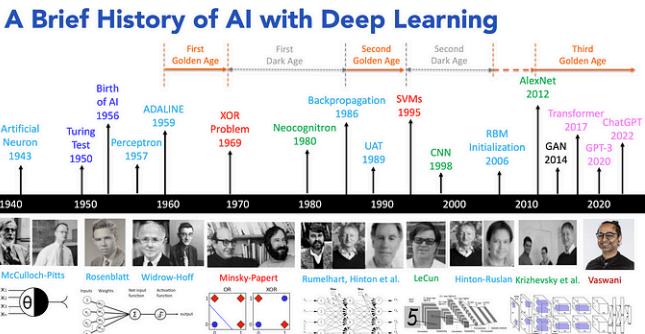


 Sanjay Basu, PhD in Physics, Philosophy & more

The Growing Role of Computational Scientists in Scientific Discovery

2024 Noble Prizes in Physics and Chemistry

★ Oct 10 🖱 73



 LM Po

A Brief History of AI with Deep Learning

Artificial intelligence (AI) and deep learning have seen remarkable progress over the pas...

★ Sep 1 🖱 196 💬 5



Lists



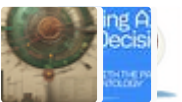
Natural Language Processing

1759 stories · 1358 saves



The New Chatbots: ChatGPT, Bard, and Beyond

12 stories · 483 saves



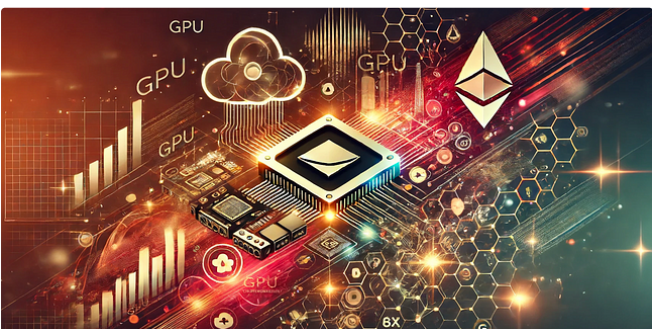
data science and AI

40 stories · 268 saves



Staff Picks

748 stories · 1376 saves



Unleashing the Power of GPUs: From On-Premises to Cloud...

Jun 14



Siddhartha Shrestha

Deploying ML Models using Nvidia Triton Inference Server

Triton Inference Server enables teams to deploy any AI model from multiple deep...

Jun 11

48



Unlocking Earth's Secrets: How Machine Learning is...

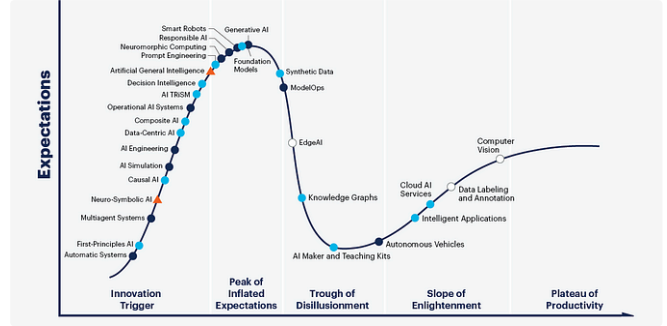
The Awakening of Data



Sep 22



136



Vishal Rajput in AIguys

Why GEN AI Boom Is Fading And What's Next?

Every technology has its hype and cool down period.



Sep 4



2.3K



71



See more recommendations