

Applied Accelerated Artificial Intelligence
Prof. Satyadhyan Chickerur
School of Computer Science and Engineering
KLE Technological University
Indian Institute of Technology, Palakkad

Lecture - 58
Applied AI: Healthcare (Federated Learning, AI Assisted Annotation)
Session I - Part - 1

Good evening everyone. So, let us start today's session which is the penultimate session second last session rather of Applied AI Healthcare related case study right. So, let me just start with the context of what we did in the previous session we actually gave the case study of something which was called as video analytics and that video analytics basically talked about how we could use it in a smart city type of a scenario.

Now, in this particular session and the next we are trying to concentrate on something which is related to health care. So, we are trying to talk of how AI can be applied to healthcare and more specifically to something like federated learning and AI assisted annotation right. So, this is the overall session thing right for the next for this session as well as the next session right.

(Refer Slide Time: 01:33)

 **Agenda**

- Federated Learning
- Distributed Learning vs Federated learning
- How it Works ?
- Tensorflow Federated
- NVIDIA FLARE
- Concept of Controllers

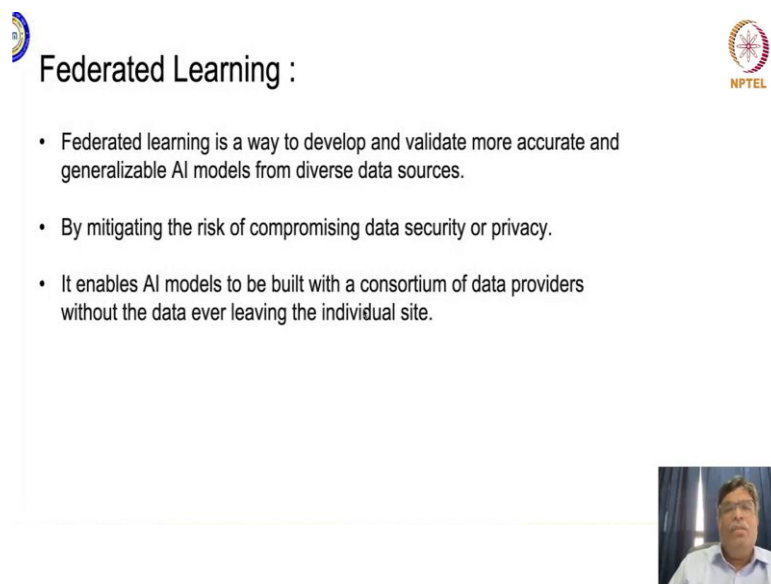


So, let us start today's session which basically would be concentrating on this, this would be today's agenda. So, we would first discuss as to what federated learning is. So, we

and what is the difference between distributed learning and federated learning. We will see how it works, we will then try to tell you in brief about something which is called as tensorflow federated may be, we will we will also tell you about something which is called as NVIDIA FLARE which also has got the concept of controllers right.

And today we will do hands on tensorflow federated which basically is in Google colab which we will be sharing it with you. So, you can actually do hands on that. So, today we will concentrate on that, tomorrow we will concentrate on NVIDIA FLARE in detail as to how it can be implemented ok. So, that will cover the total applied AI for healthcare case studies.

(Refer Slide Time: 02:34)



The slide features a title 'Federated Learning' on the left, accompanied by a small circular logo. On the right, there is the NPTEL logo. Below the title, three bullet points describe federated learning. At the bottom right of the slide, there is a small video inset showing a man speaking.

Federated Learning :

- Federated learning is a way to develop and validate more accurate and generalizable AI models from diverse data sources.
- By mitigating the risk of compromising data security or privacy.
- It enables AI models to be built with a consortium of data providers without the data ever leaving the individual site.

So, to start with let us understand what federated learning is right. So, federated learning is something like a decentralized type of a learning technically and it is basically a way to develop and validate more accurate and generalizable AI models from diverse data sources right. So, what does it mean? We will discuss in some of the other slides as to what do we mean by this when we say diverse data sources right.


So, you can have data sources from various modalities or from various geographical regions or from various hospitals ok or various types of datas ok. So, various types right and that basically is a input to a different type of a model which is not totally centralized. It is a decentralized type of a learning model right; so, distributed type of a learning model.

So, we will be discussing that and when we implement this federated learning right what is the requirement of doing such type of a decentralized concept of trying to make the models learn in such a manner. One of the things is by mitigating the risk of compromising data security or privacy right. So, this is now a very very important thing in healthcare domain as to hospitals are actually not supposed to be sharing in the patients data confidentiality clause so on and so forth and obviously, when you are trying to do research as well you do not want to share your whole data with other people ok.


So, the basic idea is that you do not want your data to be given to other people, but you are ok with that data being used ok for training some model and this model can be used by each and every one of the person who actually contributes to training that model right. So, in a sense you are not sharing the data, but you are trying to train that model with your data. So, that other people can use that particular model for their own prediction or for their own use right.

So, there is a way by which you can actually try to secure your data ok. So, this is what a federated learning actually means again and this is going to enable AI data models or AI models to be built with a consortium of data providers without the data ever leaving the individual site. So, this is what I was meaning when I told about data security or privacy in context of the model being used by the concerned people, but data is not being shared right. So, that is how the context is for federated learning ok.

(Refer Slide Time: 05:59)



Distributed Learning Vs. Federated Learning	
Distributed Learning	Federated Learning
Aims at Parallelizing Computing Power.	Aims at training on heterogeneous dataset.
Training single model on multiple servers.	No such hypotheses.
Assumptions : local datasets are independent and identically distributed & roughly have the same size.	Assumptions : datasets are typically heterogeneous and their sizes may span several magnitude.
Nodes are typically datacenters that have powerful computational capabilities and are connected to one another with fast network.	Commonly rely on less powerful communication media(i.e Wi-Fi) and battery-powered systems(i.e smart phones and IoT devices).



So, now there is a bit of a distinction between distributed learning or distributed deep learning and federated learning ok. So, when you say distributed deep learning the idea is to parallelize and use the compute power right. So, you are trying to parallelize your algorithm in such a way that your compute power is totally used ok by distributing your workload ok. So, then you have various type of parallelism that is a different issue I am not going into the details you have model parallelism you have data parallelism so on and so forth.

So, the basic idea is that there you are using parallel algorithms for distributing your work and that is basically for training purposes. So, that is why we call it as distributed deep learning. We talk of federated learning the idea is to train on heterogeneous data sets. So, the idea of heterogeneous data sets basically means they can be geographically dispersed data or they can be data from different people or they can be data from different institutions ok and you will use those data ok for training the model.

And when you talk of distributed deep learning you are training single model on multiple servers, that basically means here the concept is you have a model and you are distributing either that model or the data which you are using to train that model is distributed on various servers or various compute elements anything is possible right. So, you call it as distributed deep learning. In case of federated learning there is no such hypothesis or there is no such concept.

The basic idea is that you are going to train a model on data set which basically is not shared with other people, but the training results ok or the values obtained after training the model could be shared with others right. So, this is the basic idea sorry. So, the assumption when you are trying to work with a distributed type of a learning environment is that the local data sets are independent and identically distributed and roughly have the same size.

So, this basically means that your data set is independent and ideally distributed right. They are identical and they are distributed and they are of same size. In case of federated learning ok since we are talking of heterogeneous data sets you can get data sets which are of different sizes right. In a sense it will not be uniform ok that is what it means and when you talk of distributed data distributed deep learning you talk of nodes right nodes or the compute elements or computers or whatever you talk about right.

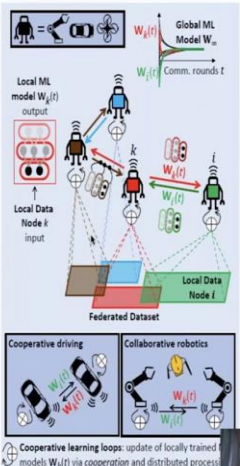
Whereas federated learning you are shifting towards something like less powerful medias right like Wi-Fi you want to use battery power systems, you will be using smart phones you will be using IoT devices on the edge ok, the something of that sort. So, you are moving from actually a data centric approach ok to something which is a inferencing type of a approach, but the essence is when you are trying to use this type of a learning you will ensure that data is not shared ok that is what is the interest ok.

(Refer Slide Time: 09:58)


Federated (decentralized) Learning

- Federated learning aims at training a machine learning algorithm on multiple local datasets contained in local nodes without explicitly exchanging data samples.
- Federated Learning trains central models on decentralized data.
- **“Federated Learning is born at the intersection of on-device AI, blockchain, and edge computing/IoT”.**

https://researchgate.net/publication/351668446_federated_learning_in_5g_nets_for_internet_of_things and <https://arxiv.org/pdf/2101.03867.pdf>



Local ML model $W_k(t)$ output
 Local Data Node k input
 Federated Dataset
 Global ML Model W_g
 Comm. rounds t
 Local Data Node l
 Cooperative driving
 Collaborative robotics
 Cooperative learning loops: update of locally trained models $W_k(t)$ via cooperation and distributed process



So, let us try to understand this. Federated learning or as I told you decentralized learning it aims at training a machine learning algorithm on multiple local data sets contained in a local nodes without explicitly exchanging data samples ok. So, this means that you are going to train a machine learning model and that machine learning model would be used for local data sets on certain local nodes without exchanging any data samples ok and federated learning train central models on decentralized data.

So, that is what it means. You are actually using a single central model it is not distributed across various nodes ok. It is a single centralized model which you are training, but you are training it under the conditions that your data is decentralized. Now why was this of importance right? This was important because you are talking of a convergence right of on device AI block chain edge computing or IoT ok.

So, you are talking of applications which are at the convergence of ok on device, AI block chain and edge computing. So, this is how the federated learning concepts started

and you are talking of decentralization of the data ok. Now let us try to understand this example. So, here if you see we are trying to actually understand that this basically is a let us say a robot right it needs to be trained. So, we are talking of collaborative robotics we are talking of cooperative driving there are so many examples, but this was just a example which will make the things easier. So, I chose this example.

So, the effective idea is that these robots right or these autonomous vehicles or whatever you talk about ok or agents or anything you talk about, these can be trained using their own right local data sets. So, this means this is being trained this also is being trained this also is being trained this also is being trained with the local data.

Now what effectively happens is all of these are trained using their local data and there will be a situation where in the information or the knowledge which this particular robot gets will be shared ok with their neighboring robots or will be shared with other what to say other users of people right.

So, you can this is what actually I was talking about that you have got this local data right, with this local data you are training your model or you are trying to train your robot or whatever with these local data ok and then all of these will share it among themselves right and then you will come up with a model which is trained because of the information from this particular model this particular model and since this also has shared some data with this.

So, this ultimately is trained and that information is already available in this model. So, this again is shared with this. So, effectively what you are trying to do is you are going to share the information about the model right. And then you are going to come up with a common global weights which is the effect of the weights from this from this from this and then you come up with a global model which is learned right and this particular data which it is being used ok is of different size different type or whatever right and this is also different this also different.

So, ultimately you come up with a federated data set which is combination of you know this and all of that right. So, this is the basic idea of what you are trying to achieve ok. This is a bit different from distributed deep learning ok.

(Refer Slide Time: 15:00)



How it works.....?



1. So, our **centralized** machine learning application will have a **local copy** on all devices, where users can use them according to their need.
2. The model will now gradually **learn and train** itself on the information provided by the user and become **smarter** from time to time.
3. The devices are then allowed to **transfer** the training results, from the local copy of the machine learning app, back to the central server.

Remember, only results, not data!

4. This same process happens across several devices, that have a local copy of the application. The results will be **aggregated** together in the centralized server, this time without user data.
5. The centralized cloud server now **updates its central machine learning model** from the aggregated training results, which is now far better than the previously deployed version.
6. The development team now updates the model to a **newer version**, and users update the application with the **smarter model**, created from their **own data!**



So, the idea is that our centralized machine learning algorithm ok will have local copy on all the devices where users can use them according to their needs right. So, that is what basically a centralized machine learning algorithm having its own local copies on all the devices. Now, the model is now gradually going to learn and train itself ok on the information provided by that particular user and it will become smarter from time to time.

So, the basic idea is it will attain some local knowledge ok and train itself. So, the basic gist is once you have some common copy which is with all of these people ok, all the models are being trained ok with some specific thing and it shared with everyone. Now, once that model comes here let us say at this i^{th} node ok it will improve its performance by getting trained with the local data which is present there on this particular node i ok.

Once it trains itself that information right it can be shared with this particular node k ok which will improve its own performance since this information also is available with this node k now right. So, this happens in a collaborative fashion right. So, that is why you are trying to talk of the essences that are model is learning ok, but the data is not getting shared right. So, that is what it means.

So, now, the devices are then allowed to transfer the training results from the local copy of the machine learning app back to the central server. Remember that we are talking

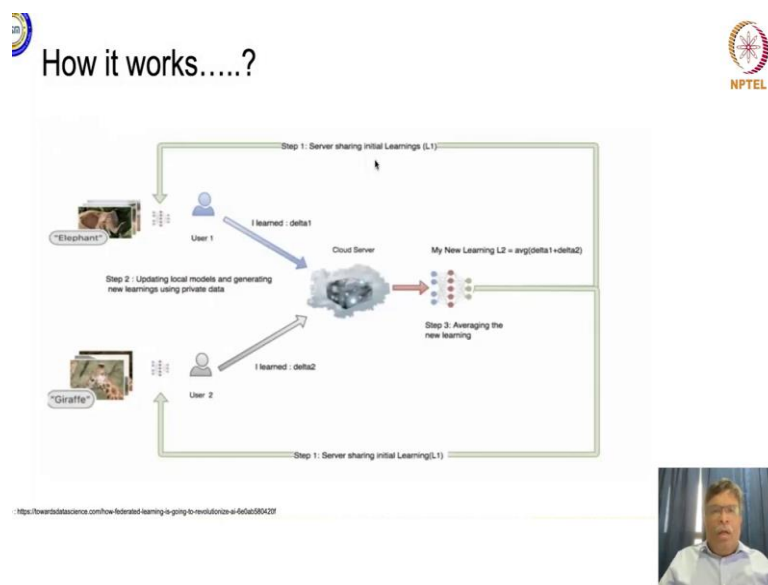
only results not the data. So, the idea is our results are getting shared ok. We are sharing the weights we are sharing various other parameters, but not data ok.

So, the same process is going to happen across all the devices which will have their own local copy of the applications, the results will be aggregated again in a centralized server ok and that again means we are not going to share the user data right. So, the centralized cloud server is now going to update its central machine learning model from these aggregated results ok, which is far better than the previously deployed version.

So, this basically means since when you would have deployed you would have deployed using only this data set right. You would have improved still you would have improved still, but the variation in datasets to improve your performance right, you will have to train your model with heterogeneous data set right.

So, you will get again this is a different heterogeneous dataset, this also again a different heterogeneous. So, this information of your model behaving on these data sets will be shared with you on this. So, your model also improves right.

(Refer Slide Time: 18:13)



So, this is how it is going to happen. So, this let us try to understand this as an example that the first step is that the server is sharing its initial learning. So, there is a centralized server assume that there is a centralized server ok and that server shares its initial learning.

So, it has a trained deep neural network and it is shared with this particular user 1 and this particular user 2 right and now there are various images of elephants here let us say ok. This is just an example which I am just trying to correlate and then there is this particular thing of let us say giraffes ok, different types of different varieties of giraffes or whatever. Now in this case what effectively happens is that now you have got a common model which is shared ok by a central server.

Now, what you are supposed to be doing is that this local data which you have which is images of elephants let us say ok. You are using it to train this model and this particular user is using these these particular images which are giraffe in this case ok to train this specific model. So, what effectively happens is you are updating your own respective local models and generate some new learning's for this model, but the data is there at your local place.

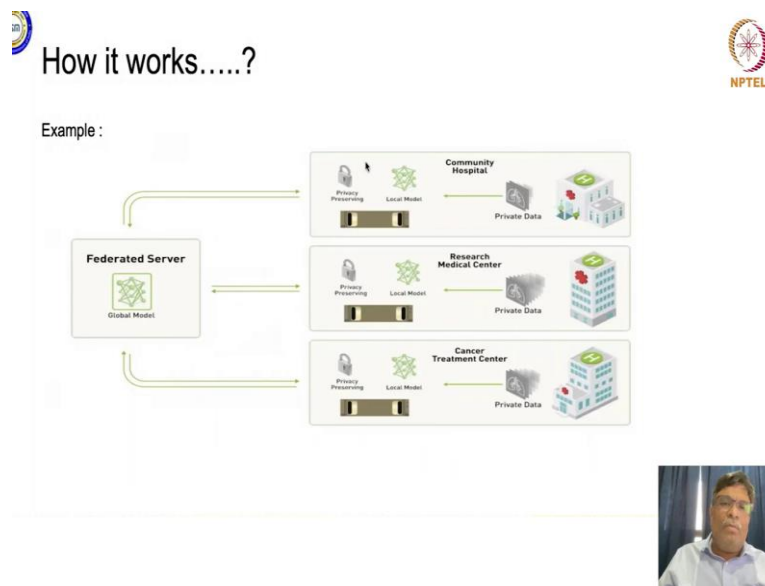
So, now you have learned something which is let us say Δx you have improved on your performance from this particular shared initial learning model and this person also would have learned something more ok because you have added this now. So, you have learned something and this let us assume its $\Delta 2$. So, you have learned something more you have learned something more.

Now, you share that in such a way that the model which you have developed is going to improve ok, but what is the understanding or a general standard procedure is that your new learning of 1 2 would be the average of what is there in $\Delta 1$ and $\Delta 2$ right. This is generally called as averaging the new learning or whatever you talk about right. So, this is how it goes on improving the model.

So, now if there are 10 such clients right which have their own local data ok. You basically learn some local information ok and you basically get Δx and Δy and you add this Δx plus Δy average it out and this model improves a bit more and this again will be shared with everyone ok. So, this is how it is going to happen. So, this is what is basically federated learning right.

I hope this is clear to everyone as to what are you going to do in this right. This is a conceptual thing of basically first understanding and then how do you implement is a different issue we will tell you that ok, but this is what the concept says the now ok.

(Refer Slide Time: 21:42)



In the context of health care data right. So, how is it going to actually work? So, let us say there is this global federated server or there is a central server this is to be assumed as a global model or this is a global model. Now, we have got hospital 1 hospital 2 hospital 3. So, these are of various categories right. You can assume it to be a community hospital this can be a research center this can be a treatment center or whatever.

Each of these centers will have their own private data and sharing this private data with everyone right is out of question. So, what you do in such a scenario is that you want a good AI model ok to be developed. And since it needs to be developed the inferencing from this data model alone ok would be better when you combine the learning of this model and this model.

So, what do you want to do? You want to combine and get a global model which is effectively a sum total of the learnability which has happened to this model plus this model plus this model and these models have learned and you can use that learnability to come in to come up with a global model wherein absolutely the data is not shared.

So, once this is updated, this global model can be used by this community hospital as well as this research center as well as this treatment center to come up with their own prediction for their input data and again as and when the data goes on accumulating here this community hospital again fine tunes this model which is locally done ok.

This also does the same thing, this also does the same thing and ultimately you again share the learning here by preserving the privacy and then update this global model. So, that it could be used by all the other stakeholders. So, this is how it is going to actually work right.