Enterprise Challenges in Federated AI Solutions

—

Dinesh C. Verma, IBM Fellow,
CTO Edge Computing, IBM Research
Email: dverma@us.ibm.com

This research was sponsored in part by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Enterprise Federated Learning: Introduction

Machine Learning model quality influenced strongly by training data that is available

Ideally, training data would be available at a central location

– Central location has sufficient compute with GPU/FPGA assists and good local network connectivity

Unfortunately, many situations are far from ideal

– We consider situations where training data is distributed across many locations

Typical steps in AI enabled System

Preparation

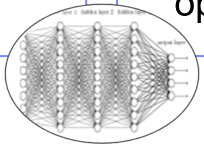
- Collect training data

Learning

- Train the AI model

Inference

- Use trained model in operation



Federated Learning: Introduction

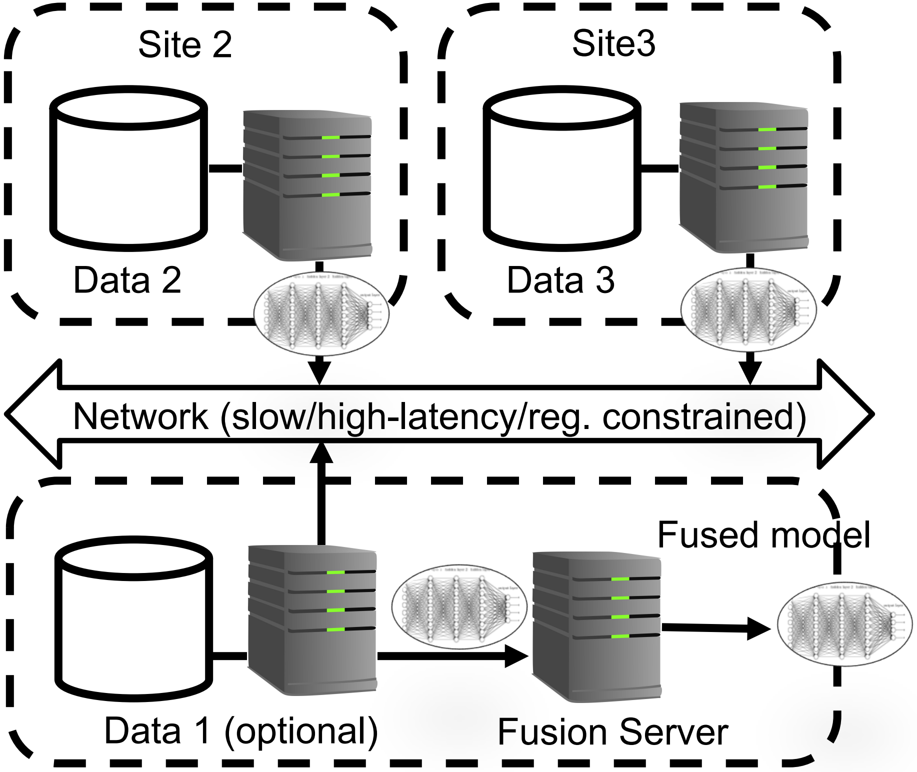
Training data is distributed at many sites

- Data can not be moved into a single site
- Each site has capability to train local models
- Some level of coordination is permitted
  - Server based coordination

Basic Approach:

- Each site trains its own model
- Server site helps to fuse models from different sites

Definition: Federated Learning is the technology to let models be trained on widely distributed data sets, and combine the models to produce one equivalent to one produced by centralized training.



Consumer vrs Enterprise Federated Learning

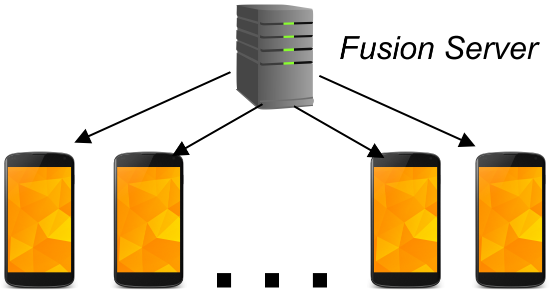
Our Focus

Consumer Federated Learning

Homogenous data

Split in large numbers (thousands to millions)

Learning on mobile phones/consumer devices



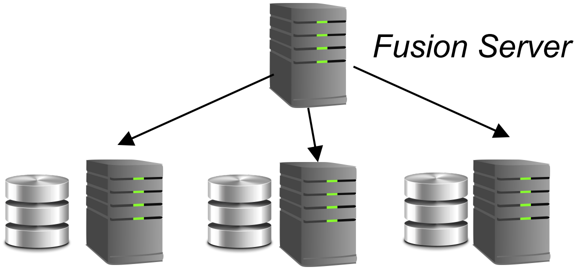
Millions of phones with small parts of data
All phones run the same app (same data format)

Enterprise Federated Learning

Heterogenous data & perhaps different models

Data Split in relatively small number of sites

Learning on large servers or data centers



Handful of sites with large volumes of data
Data may have different format, quality & constraints

Consumer vrs Enterprise Federated Learning

Consumer Federated Learning

Motivation

– Privacy of consumer data on phone

Challenges

– Small amounts of data per participant

– Guaranteeing privacy of information

– Malicious participants

Simplifications

– Same data format/schema

– Synchronization ease

– law of of large numbers

Our Focus

Enterprise Federated Learning

Motivation

– Cost of data movement, regulatory concerns

Challenges

– Differences in data schematics and quality

– Synchronization difficulties

– Sites may contain different functions

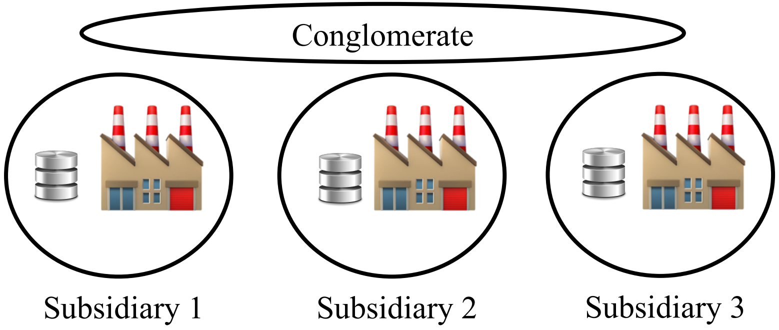
Simplifications

– Business arrangements for trust/security

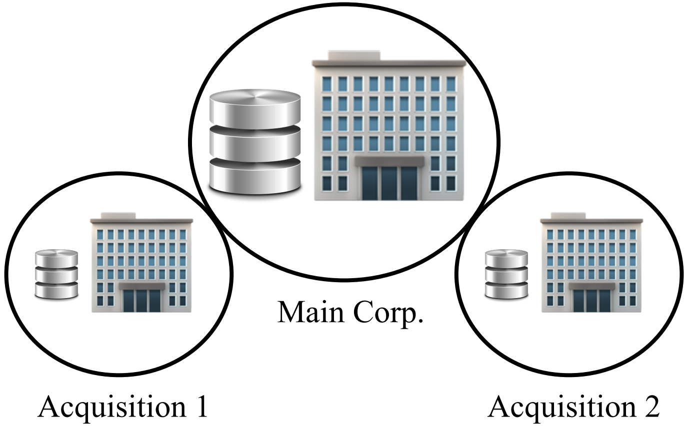
– Enough data to train a good model at each site

Enterprise Federated Learning Scenarios

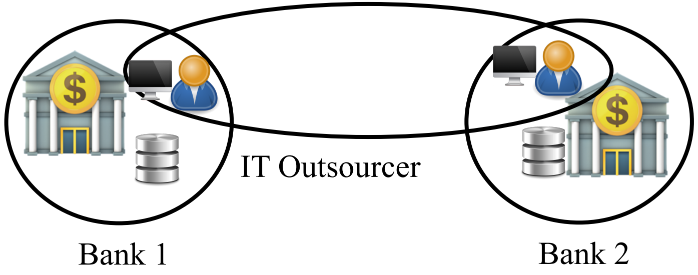
Mergers & Acquisitions



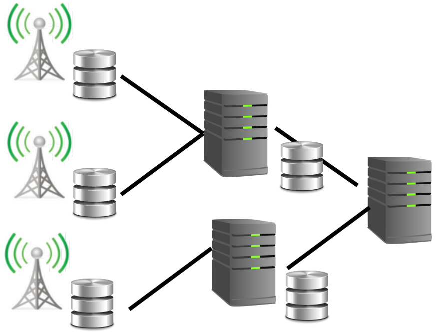
Subsidiaries & Franchises



Outsourced Operations

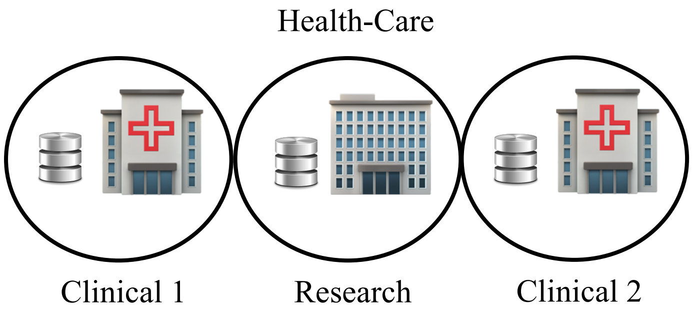


Operations at Scale

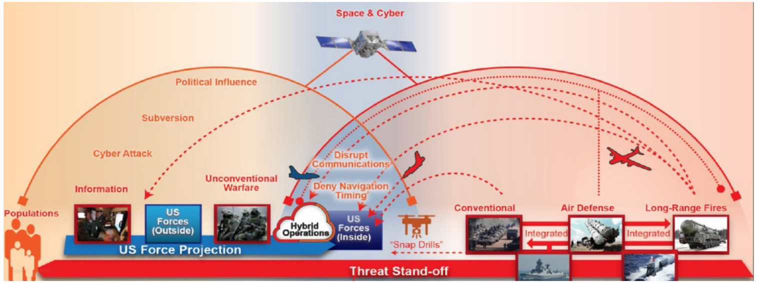


Enterprise Federated Learning Scenarios (contd)

Regulated Industries

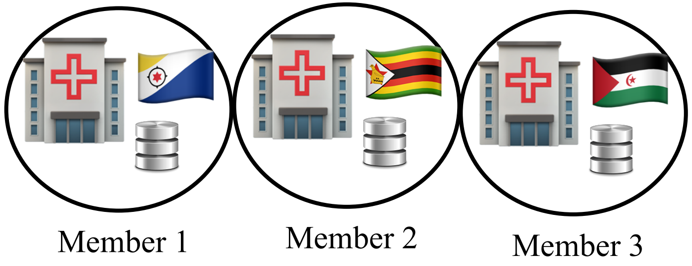


Multi-Domain Operations

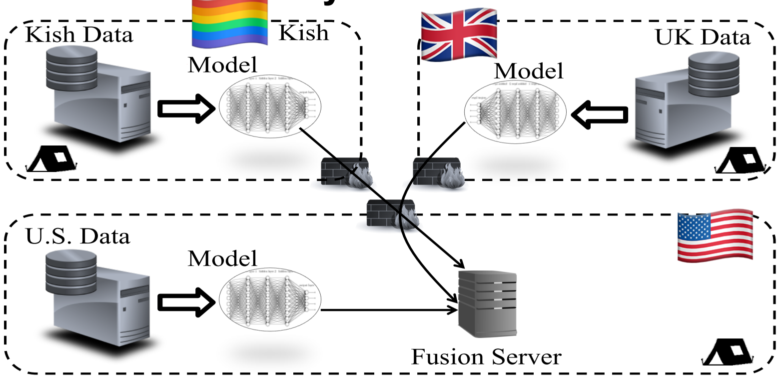


Source: TRADOC Pamphlet 325-3-1: The U.S. Army in Multi-Domain Operations 2028

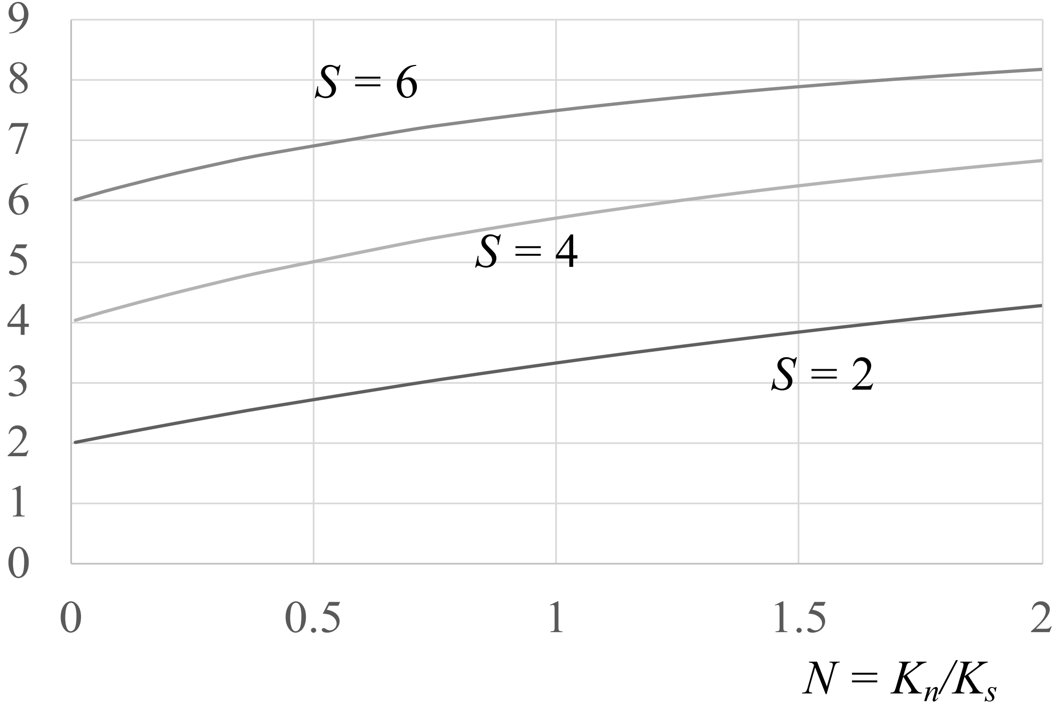
Consortiums



Military Coalitions



Comparing FL time taken versus Moving Data Centrally



K_n -- time to transfer a given size of data across the network

K_s – time to train a machine learning model on the same size of data.

$N = K_n/K_s$ the relative performance of network to compute

S - number of sites involved

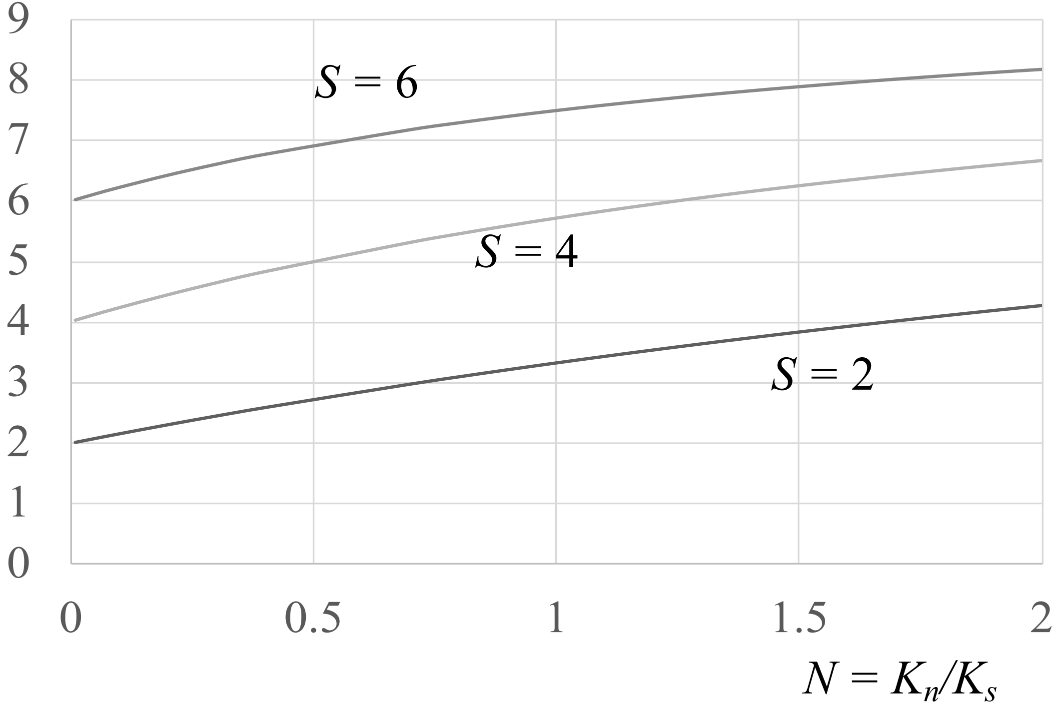
M_r - model reduction ratio, size of models transferred compared to training data transferred

Relative Time for Training = $(1+ N)/(1/S + NM_r)$

Asymptotic limits:

- N → inf; speedup = $1/ M_r$
- N → 0; speedup = S
- S → inf; speedup = $(1+1/N)/M_r$

Enterprise Scenario: Federated Learning is usually faster
Consumer Scenario: Unclear outcome



K_n -- time to transfer a given size of data across the network

K_s – time to train a machine learning model on the same size of data.

$N = K_n/K_s$ the relative performance of network to compute

S - number of sites involved

M_r - model reduction ratio, size of models transferred compared to training data transferred

Relative Time for Training = $(1+ N)/(1/S + NM_r)$

Asymptotic limits:

- N → inf; speedup = $1/ M_r$
- N → 0; speedup = S
- S → inf; speedup = $(1+1/N)/M_r$

Complexities

Assumptions of Averaging

– Data is similar across all sites
– Data schema is same at all sites
– All classes present at all sites (for classification)
– Same function learnt at each site
– All sites are training at the same time
– Number of sites does not change

Reality in Enterprises

– Issues which prevent data movement also result in data format and data ranges being different
– Different classes present at different sites
  • Classes named differently
– Different functions at different site
  • Catastrophic forgetting in neural networks
– Sites can not synchronize easily and train at same time
– Sites may change over time

Complexities

Assumptions of Averaging

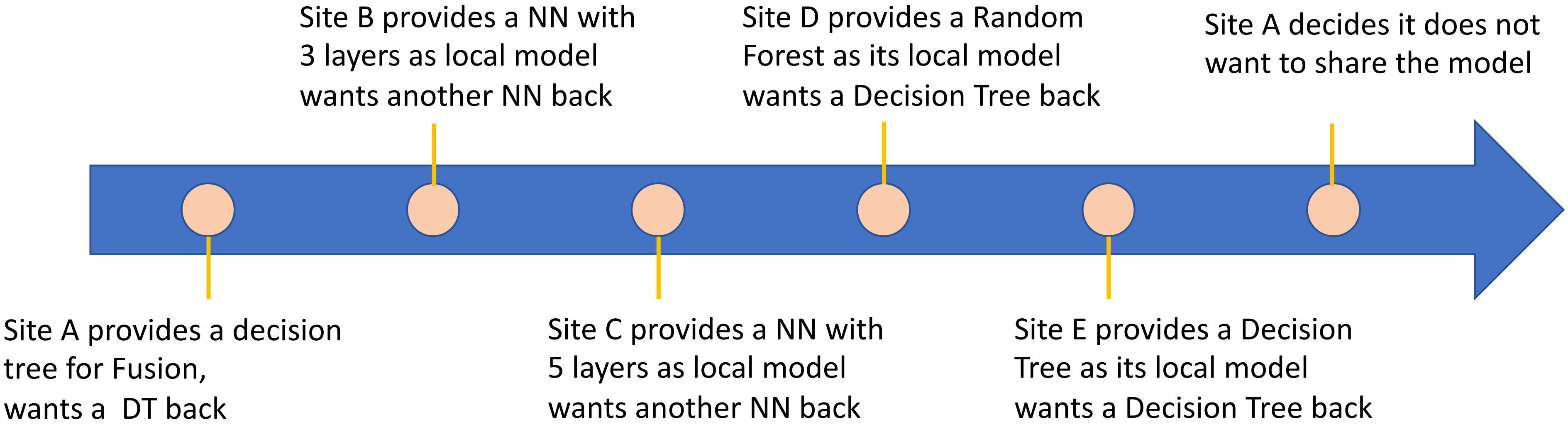
- Data is similar across all sites
- Data schema is same at all sites
- All classes present at all sites (for classification)
- Same function learnt at each site
- All sites are training at the same time
- Number of sites does not change

Reality in Enterprises

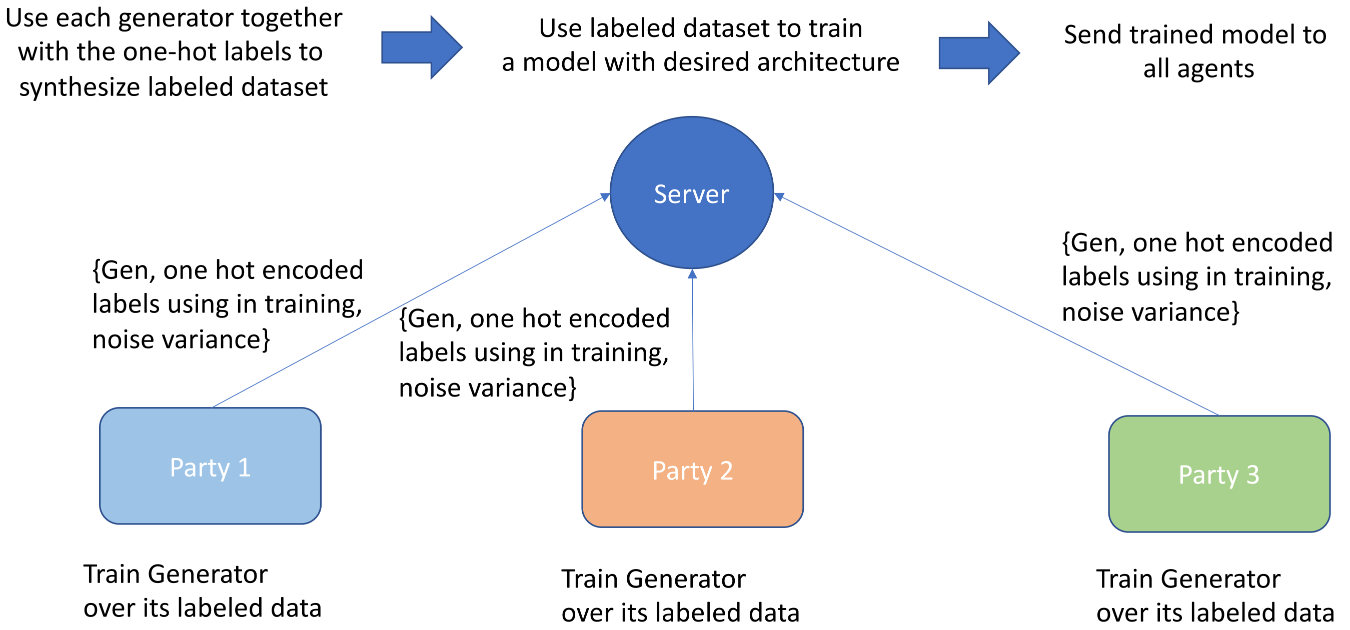
- Issues which prevent data movement also result in data format and data ranges being different
- Different classes present at different sites
  - Classes named differently
- Different functions at different site
  - Catastrophic forgetting in neural networks
- Sites can not synchronize easily and train at same time
- Sites may change over time

Challenge: Synchronization

- Support a scenario that unfolds in this manner, with the fusion site already having a fused model which is a 10-layer neural network.



Solution: One Shot Federation



Type of Generators

- Conditional GANs
- Stochastic Modelers
- Neural Embeddings + Distribution

- Generators + Classifiers

Type of data determines the best type of generator

Effectiveness of Eventual Models depends on type of generator used

Implications for Systems

Systems support needed at all site for converting data to a generator models

- Fast generator models
- Fast matrix manipulations for PCA transformations
- Fast statistics computation
- Fast generator model training (similar to existing ML requirement)

Systems requirement for other enterprise challenges

- Fast scaling of data to a canonical format
- Fast embedding based matching for class name resolution

Systems support needed to generate representative data

- Fast random number generators
- Fast extraction of distributed number generators

Fast Training of Models (existing ML requirements)

For more details

Various papers published on federated learning by IBM Research Colleagues

https://www.amazon.com/dp/B099F6VG2Q/