Chap3#5: Federated & Ensemble Learning in Machine Learning & Case study of Privacy Preserving ML Applications

March 26, 2023

Devesh C Jinwala,
Professor, SVNIT and Adjunct Prof., CSE, IIT Jammu

Department of Computer Science and Engineering,
Sardar Vallabhhai National Institute of Technology, SURAT

Chap 3: Privacy Preserving Machine Learning : Topics to study

- Privacy Preservation, What is Privacy? Data Privacy. Machine Learning in Privacy Preservation: Four Main stakes to Privacy preservation in ML. Two principle approaches: (a) Augmenting the ML techniques with the conventional approaches in the domain of privacy preservation to achieve privacy viz. Homomorphic Encryption, Secret Multiparty Computations, Zero Knowledge Proofs, Perturbation techniques (e.g. differential privacy) Anonymization techniques (e.g.)k-Anonymity, l-Diversity) (b) ML-specific approaches like Federated Learning OR Ensemble Learning. Homomorphic Encryption Algorithms and the associated mathematics. Ethical issues and Law for data / process privacy : GDPR, Alexa, other relevant applications [6 hours]

Chap3#5:Federated & Ensemble Learning for Privacy Preservation & Applications

Four Main stakes to Privacy preservation in ML

There are four main stakes to privacy preservation in general:

- Privacy of the input data
  - the assurance that other parties, including the model developer, will not be able to see a user's input data
- Privacy of the output data
  - the assurance that the output of a model is only accessible to the client whose data is being inferred upon.
- Privacy of the model
  - rhe assurance that a hostile party will not be able to steal the model
- Data privacy in training
  - the assurance that a malicious party will not reverse-engineer the training data - although gathering information about training data and model is more difficult than that for the data.

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like
    - homomorphic encryption

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like
    - homomorphic encryption
    - secure multi-party computing,

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like
    - homomorphic encryption
    - secure multi-party computing,
    - Zero knowledge proofs

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like
    - homomorphic encryption
    - secure multi-party computing,
    - Zero knowledge proofs
  - perturbation techniques like differential privacy

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like
    - homomorphic encryption
    - secure multi-party computing,
    - Zero knowledge proofs
  - perturbation techniques like differential privacy
  - anonymization techniques like k-Anonymity and l-Diversity

Privacy Preserving Machine Learning: How to achieve?

The goal of privacy-preserving machine learning is

- to bridge the gap between privacy while receiving the benefits of machine learning.
- is a critical facilitator for the protection of acquired data and adhering to data privacy laws.

Privacy-preservation in ML

- is achieved by augmenting conventional ML with different strategies that protect data privacy, that include....
  - cryptographic approaches like
    - homomorphic encryption
    - secure multi-party computing,
    - Zero knowledge proofs
  - perturbation techniques like differential privacy
  - anonymization techniques like k-Anonymity and l-Diversity
  - ML-specific approaches like Federated Learning OR Ensemble Learning - the Privacy-Preserving Techniques - modifying the conventional ML training methods to keep user data private.

ML specific approaches for Privacy Preservation: Federated Learning

PPML : Federated Learning, Motivation

- Quality data exist as islands on edge devices like mobile phones and personal computers across the globe

PPML : Federated Learning, Motivation

- Quality data exist as islands on edge devices like mobile phones and personal computers across the globe
- these devices and therefore the data are guarded by strict privacy preserving laws.

PPML : Federated Learning, Motivation

- Quality data exist as islands on edge devices like mobile phones and personal computers across the globe
- these devices and therefore the data are guarded by strict privacy preserving laws.
- How to use such data for an ML model ?

PPML : Federated Learning, Motivation

- Quality data exist as islands on edge devices like mobile phones and personal computers across the globe
- these devices and therefore the data are guarded by strict privacy preserving laws.
- How to use such data for an ML model ?
- Federated Learning is driven by an attempt to answer such questions.

PPML : Federated Learning, Motivation

- Quality data exist as islands on edge devices like mobile phones and personal computers across the globe
- these devices and therefore the data are guarded by strict privacy preserving laws.
- How to use such data for an ML model ?
- Federated Learning is driven by an attempt to answer such questions.

PPML : Federated Learning, Motivation

- Quality data exist as islands on edge devices like mobile phones and personal computers across the globe
- these devices and therefore the data are guarded by strict privacy preserving laws.
- How to use such data for an ML model ?
- Federated Learning is driven by an attempt to answer such questions.

Federated Learning (FL)

- provides a clever means of connecting ML models to these disjointed data regardless of their locations

PPML : Federated Learning, Motivation

- Quality data exist as islands on edge devices like mobile phones and personal computers across the globe
- these devices and therefore the data are guarded by strict privacy preserving laws.
- How to use such data for an ML model ?
- Federated Learning is driven by an attempt to answer such questions.

Federated Learning (FL)

- provides a clever means of connecting ML models to these disjointed data regardless of their locations
- more importantly, it does so without breaching privacy laws.

PPML : Federated Learning, Motivation

- Quality data exist as islands on edge devices like mobile phones and personal computers across the globe
- these devices and therefore the data are guarded by strict privacy preserving laws.
- How to use such data for an ML model ?
- Federated Learning is driven by an attempt to answer such questions.

Federated Learning (FL)

- provides a clever means of connecting ML models to these disjointed data regardless of their locations
- more importantly, it does so without breaching privacy laws.
- it is sort of using the principle that instead of taking the data to the model for training, FL takes the model to the data instead.

PPML : Federated Learning, Motivation

- Quality data exist as islands on edge devices like mobile phones and personal computers across the globe
- these devices and therefore the data are guarded by strict privacy preserving laws.
- How to use such data for an ML model ?
- Federated Learning is driven by an attempt to answer such questions.

Federated Learning (FL)

- provides a clever means of connecting ML models to these disjointed data regardless of their locations
- more importantly, it does so without breaching privacy laws.
- it is sort of using the principle that instead of taking the data to the model for training, FL takes the model to the data instead.
- thus, allows ML processes to be decentralized, lowering the amount of information exposed from contributor datasets

Federated learning

- is aimed at reducing the danger of data being leaked and prevents identity privacy being compromised.

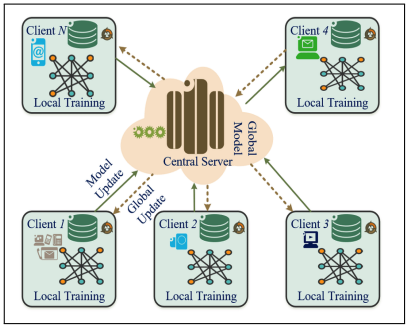

Figure: Privacy Preserving FL

Src: Unknown

Federated learning

- is aimed at reducing the danger of data being leaked and prevents identity privacy being compromised.
  - this is unlike learning in conventional ML algorithms wherein, the complete training process is carried out on a single (cloud) server.

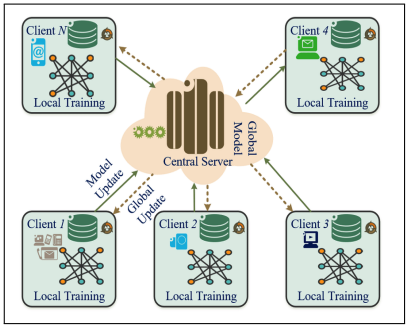

Figure: Privacy Preserving FL

Src: Unknown

Federated learning

- is aimed at reducing the danger of data being leaked and prevents identity privacy being compromised.
  - this is unlike learning in conventional ML algorithms wherein, the complete training process is carried out on a single (cloud) server.
  - when the information is shared with the central cloud server, there is a privacy risk.

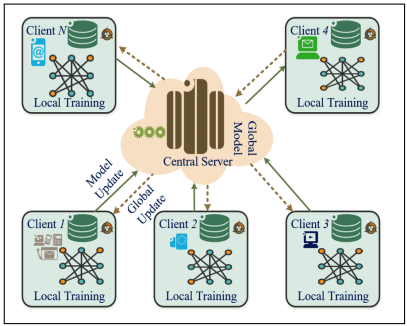

Figure: Privacy Preserving FL

Src: Unknown

Federated learning

- is aimed at reducing the danger of data being leaked and prevents identity privacy being compromised.
  - this is unlike learning in conventional ML algorithms wherein, the complete training process is carried out on a single (cloud) server.
  - when the information is shared with the central cloud server, there is a privacy risk.
- enables participants to train local models cooperatively on local data without disclosing sensitive data to a central cloud server.

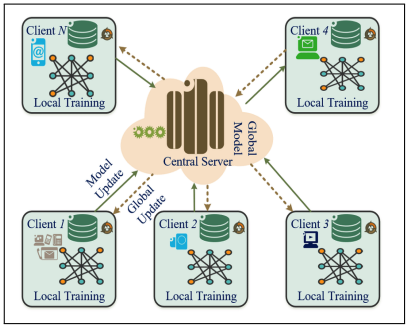

Figure: Privacy Preserving FL

Src: Unknown

Federated learning

- is aimed at reducing the danger of data being leaked and prevents identity privacy being compromised.
  - this is unlike learning in conventional ML algorithms wherein, the complete training process is carried out on a single (cloud) server.
  - when the information is shared with the central cloud server, there is a privacy risk.
- enables participants to train local models cooperatively on local data without disclosing sensitive data to a central cloud server.
- also allows continuous learning on end-user devices while ensuring no end-user data leaves the device.

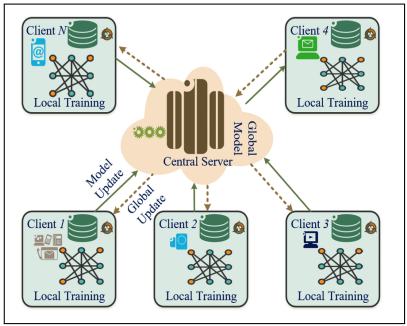

Figure: Privacy Preserving FL

Src: Unknown

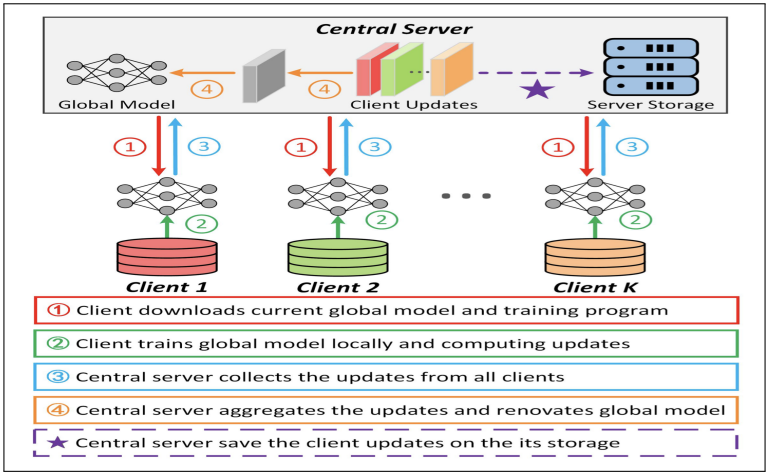

Figure: Privacy Preserving FL

[1] Source: Liu, Gaoyang et al. (Book) Federated Unlearning, arXiv, 2020

FL

- was proposed by Google in McMahan et al[1] as a promising solution that can train a unified DL model across multiple decentralized clients holding local data samples

[1] B. McMahan et al "Communication-efficient learning of deep networks from decentralized data," in Proceedings of AISTATS, 2017, pp. 1273–1282.

FL

- was proposed by Google in McMahan et al[1] as a promising solution that can train a unified DL model across multiple decentralized clients holding local data samples
- this is done under the coordination of a central server.

[1] B. McMahan et al "Communication-efficient learning of deep networks from decentralized data," in Proceedings of AISTATS, 2017, pp. 1273–1282.

FL

- was proposed by Google in McMahan et al[1] as a promising solution that can train a unified DL model across multiple decentralized clients holding local data samples
- this is done under the coordination of a central server.
- each client's data is stored on its local storage and not transferred to other clients or the central server.

[1] B. McMahan et al "Communication-efficient learning of deep networks from decentralized data," in Proceedings of AISTATS, 2017, pp. 1273–1282.

FL

- was proposed by Google in McMahan et al[1] as a promising solution that can train a unified DL model across multiple decentralized clients holding local data samples
- this is done under the coordination of a central server.
- each client's data is stored on its local storage and not transferred to other clients or the central server.
- thus, leverages the focused updates (i.e., model gradients or weights) for aggregation to construct the unified global model.

[1] B. McMahan et al "Communication-efficient learning of deep networks from decentralized data," in Proceedings of AISTATS, 2017, pp. 1273–1282.

FL

- was proposed by Google in McMahan et al[1] as a promising solution that can train a unified DL model across multiple decentralized clients holding local data samples
- this is done under the coordination of a central server.
- each client's data is stored on its local storage and not transferred to other clients or the central server.
- thus, leverages the focused updates (i.e., model gradients or weights) for aggregation to construct the unified global model.
- thus, embodies the following principles

[1] B. McMahan et al "Communication-efficient learning of deep networks from decentralized data," in Proceedings of AISTATS, 2017, pp. 1273–1282.

FL

- was proposed by Google in McMahan et al[1] as a promising solution that can train a unified DL model across multiple decentralized clients holding local data samples
- this is done under the coordination of a central server.
- each client's data is stored on its local storage and not transferred to other clients or the central server.
- thus, leverages the focused updates (i.e., model gradients or weights) for aggregation to construct the unified global model.
- thus, embodies the following principles
  - focused collection and data minimization,

[1] B. McMahan et al "Communication-efficient learning of deep networks from decentralized data," in Proceedings of AISTATS, 2017, pp. 1273–1282.

FL

- was proposed by Google in McMahan et al[1] as a promising solution that can train a unified DL model across multiple decentralized clients holding local data samples
- this is done under the coordination of a central server.
- each client's data is stored on its local storage and not transferred to other clients or the central server.
- thus, leverages the focused updates (i.e., model gradients or weights) for aggregation to construct the unified global model.
- thus, embodies the following principles
  - focused collection and data minimization,
  - mitigation of many of the systemic privacy risks resulting from traditional, centralized approaches.

[1] B. McMahan et al "Communication-efficient learning of deep networks from decentralized data," in Proceedings of AISTATS, 2017, pp. 1273–1282.

PPML : Federated Learning, An Overview...

Federated learning - thus, the basic idea can be summarized as that

- a central ML model M owned by a central authority (e.g., a company) can be further trained on new private datasets from data contributors

PPML : Federated Learning, An Overview...

Federated learning - thus, the basic idea can be summarized as that

- a central ML model M owned by a central authority (e.g., a company) can be further trained on new private datasets from data contributors
- this is done by having each contributor train locally with its dataset and then updating the central model M (i.e., updating the model's parameter). For this purpose,

Federated learning - thus, the basic idea can be summarized as that

- a central ML model M owned by a central authority (e.g., a company) can be further trained on new private datasets from data contributors
- this is done by having each contributor train locally with its dataset and then updating the central model M (i.e., updating the model's parameter). For this purpose,
  - a group of n participants (data contributors) receives the central model M;

Federated learning - thus, the basic idea can be summarized as that

- a central ML model M owned by a central authority (e.g., a company) can be further trained on new private datasets from data contributors
- this is done by having each contributor train locally with its dataset and then updating the central model M (i.e., updating the model's parameter). For this purpose,
  - a group of n participants (data contributors) receives the central model M;
  - each participant updates the model M locally by training it on its own local dataset Z_l, yielding a new local parameter l.

Federated learning - thus, the basic idea can be summarized as that

- a central ML model M owned by a central authority (e.g., a company) can be further trained on new private datasets from data contributors
- this is done by having each contributor train locally with its dataset and then updating the central model M (i.e., updating the model's parameter). For this purpose,
  - a group of n participants (data contributors) receives the central model M;
  - each participant updates the model M locally by training it on its own local dataset Z_l, yielding a new local parameter l.
  - the central authority receives each participant's update l;

Federated learning - thus, the basic idea can be summarized as that

- a central ML model M owned by a central authority (e.g., a company) can be further trained on new private datasets from data contributors
- this is done by having each contributor train locally with its dataset and then updating the central model M (i.e., updating the model's parameter). For this purpose,
  - a group of n participants (data contributors) receives the central model M;
  - each participant updates the model M locally by training it on its own local dataset Z_l, yielding a new local parameter l.
  - the central authority receives each participant's update l;
  - the central authority combines each participant's local parameters to form a new parameter that is used to update the central model.

PPML : Federated Learning, An Overview...

Federated learning - thus, the basic idea can be summarized as that

- a central ML model M owned by a central authority (e.g., a company) can be further trained on new private datasets from data contributors
- this is done by having each contributor train locally with its dataset and then updating the central model M (i.e., updating the model's parameter). For this purpose,
  - a group of n participants (data contributors) receives the central model M;
  - each participant updates the model M locally by training it on its own local dataset Z_l, yielding a new local parameter l.
  - the central authority receives each participant's update l;
  - the central authority combines each participant's local parameters to form a new parameter that is used to update the central model.
- This procedure can be continued until the primary model is well-trained.

Federated learning yields the following benefits:

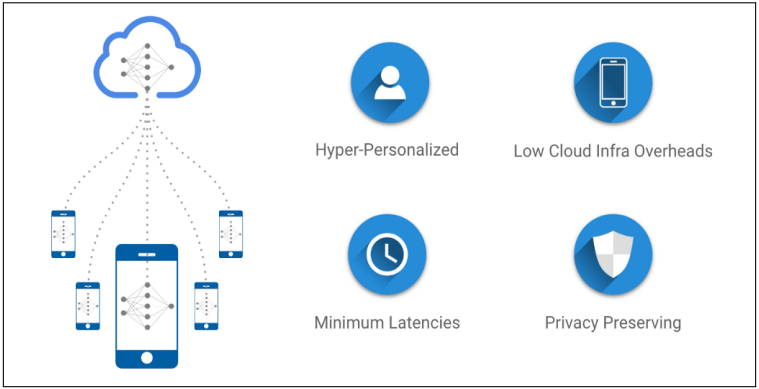

Figure: Privacy Preserving Federated Learning

1

[1] Src: GitHub

Federated learning yields the following benefits:

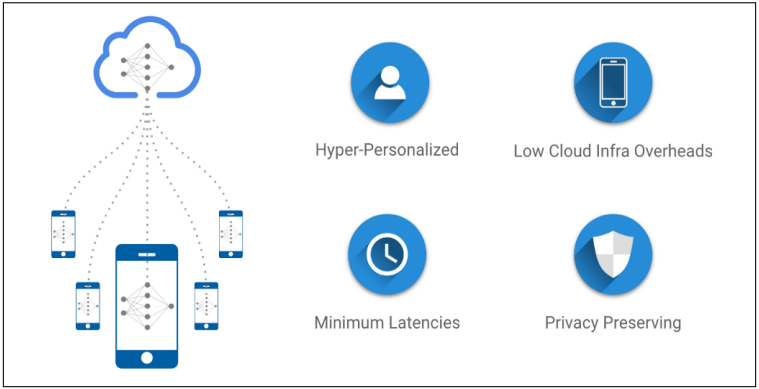

Figure: Privacy Preserving Federated Learning

Therefore, FL is often termed as the decentralized form[1] of Machine Learning.

[1] Src: GitHub

Precursors to Federated Learning

Federated Learning: Distributed Machine Learning

Distributed Machine Learning

- is a kind of multi-node machine learning designed to improve the performance, increase the accuracy, and scale the data to a large amount easily.

Federated Learning: Distributed Machine Learning

Distributed Machine Learning

- is a kind of multi-node machine learning designed to improve the performance, increase the accuracy, and scale the data to a large amount easily.
- the basic motivation is driven from the need to address two issues, minimally:

Federated Learning: Distributed Machine Learning

Distributed Machine Learning

- is a kind of multi-node machine learning designed to improve the performance, increase the accuracy, and scale the data to a large amount easily.
- the basic motivation is driven from the need to address two issues, minimally:
  - to handle massive data volume, such as societal-scale social graphs with up to hundreds of millions of nodes; and

Federated Learning: Distributed Machine Learning

Distributed Machine Learning

- is a kind of multi-node machine learning designed to improve the performance, increase the accuracy, and scale the data to a large amount easily.
- the basic motivation is driven from the need to address two issues, minimally:
  - to handle massive data volume, such as societal-scale social graphs with up to hundreds of millions of nodes; and
  - to handle massive model size, (e.g. Google Brain deep neural network) containing billions of parameters.

Federated Learning: Distributed Machine Learning

Distributed Machine Learning

- is a kind of multi-node machine learning designed to improve the performance, increase the accuracy, and scale the data to a large amount easily.
- the basic motivation is driven from the need to address two issues, minimally:
  - to handle massive data volume, such as societal-scale social graphs with up to hundreds of millions of nodes; and
  - to handle massive model size, (e.g. Google Brain deep neural network) containing billions of parameters.
- Various variations of Distributed Machine Learning architecture have been proposed as sbown on the next slide.....

Federated Learning: Distributed Machine Learning

Distributed Machine Learning

- is a kind of multi-node machine learning designed to improve the performance, increase the accuracy, and scale the data to a large amount easily.
- the basic motivation is driven from the need to address two issues, minimally:
  - to handle massive data volume, such as societal-scale social graphs with up to hundreds of millions of nodes; and
  - to handle massive model size, (e.g. Google Brain deep neural network) containing billions of parameters.
- Various variations of Distributed Machine Learning architecture have been proposed as sbown on the next slide.....
- But none of those attempts address privacy concerns.

Federated Learning: Distributed Machine Learning...

Various variations of Distributed Machine Learning architecture have been proposed:

- a distributed machine learning framework was proposed Qirong Ho, James Cipar, et al in NIPS 2013, with an objective to handle training in a massive data volume and massive model size.

Federated Learning: Distributed Machine Learning...

Various variations of Distributed Machine Learning architecture have been proposed:

- a distributed machine learning framework was proposed Qirong Ho, James Cipar, et al in NIPS 2013, with an objective to handle training in a massive data volume and massive model size.
- Xing et al.proposed in 2015, a general framework for solving the data and model parallel challenges systematically for large-scale machine learning.

Federated Learning: Distributed Machine Learning...

Various variations of Distributed Machine Learning architecture have been proposed:

- a distributed machine learning framework was proposed Qirong Ho, James Cipar, et al in NIPS 2013, with an objective to handle training in a massive data volume and massive model size.
- Xing et al.proposed in 2015, a general framework for solving the data and model parallel challenges systematically for large-scale machine learning.
- Xie et al. proposed an effective factor broadcast (SFB) calculation model, which is effective and efficient in distributed learning of a large matrix parameterized model.

Federated Learning: Distributed Machine Learning...

Various variations of Distributed Machine Learning architecture have been proposed:

- a distributed machine learning framework was proposed Qirong Ho, James Cipar, et al in NIPS 2013, with an objective to handle training in a massive data volume and massive model size.
- Xing et al.proposed in 2015, a general framework for solving the data and model parallel challenges systematically for large-scale machine learning.
- Xie et al. proposed an effective factor broadcast (SFB) calculation model, which is effective and efficient in distributed learning of a large matrix parameterized model.
- Wei et al. maximized the efficiency of network communication under a given network bandwidth among machines, to minimize parallel errors while ensuring the theoretical fusion for large-scale data parallel machine learning applications.

Federated Learning: Distributed Machine Learning...

Various variations of Distributed Machine Learning architecture have been proposed:

- a distributed machine learning framework was proposed Qirong Ho, James Cipar, et al in NIPS 2013, with an objective to handle training in a massive data volume and massive model size.
- Xing et al.proposed in 2015, a general framework for solving the data and model parallel challenges systematically for large-scale machine learning.
- Xie et al. proposed an effective factor broadcast (SFB) calculation model, which is effective and efficient in distributed learning of a large matrix parameterized model.
- Wei et al. maximized the efficiency of network communication under a given network bandwidth among machines, to minimize parallel errors while ensuring the theoretical fusion for large-scale data parallel machine learning applications.
- Kim et al. proposed a distributed framework STRADS, which optimized the throughput for classical distributed machine learning algorithms.

Federated Learning: Distributed Machine Learning...

Various variations of Distributed Machine Learning architecture have been proposed:

- a distributed machine learning framework was proposed Qirong Ho, James Cipar, et al in NIPS 2013, with an objective to handle training in a massive data volume and massive model size.
- Xing et al.proposed in 2015, a general framework for solving the data and model parallel challenges systematically for large-scale machine learning.
- Xie et al. proposed an effective factor broadcast (SFB) calculation model, which is effective and efficient in distributed learning of a large matrix parameterized model.
- Wei et al. maximized the efficiency of network communication under a given network bandwidth among machines, to minimize parallel errors while ensuring the theoretical fusion for large-scale data parallel machine learning applications.
- Kim et al. proposed a distributed framework STRADS, which optimized the throughput for classical distributed machine learning algorithms.
- Jeffrey et al. proposed in 2012, Google's first-generation deep learning system Disbelief, and split a model into 32 nodes for calculation.

Federated Learning: Distributed Machine Learning...

Various variations of Distributed Machine Learning architecture have been proposed:

- a distributed machine learning framework was proposed Qirong Ho, James Cipar, et al in NIPS 2013, with an objective to handle training in a massive data volume and massive model size.
- Xing et al.proposed in 2015, a general framework for solving the data and model parallel challenges systematically for large-scale machine learning.
- Xie et al. proposed an effective factor broadcast (SFB) calculation model, which is effective and efficient in distributed learning of a large matrix parameterized model.
- Wei et al. maximized the efficiency of network communication under a given network bandwidth among machines, to minimize parallel errors while ensuring the theoretical fusion for large-scale data parallel machine learning applications.
- Kim et al. proposed a distributed framework STRADS, which optimized the throughput for classical distributed machine learning algorithms.
- Jeffrey et al. proposed in 2012, Google's first-generation deep learning system Disbelief, and split a model into 32 nodes for calculation.
- In 2013, data and model parallelism in distributed machine learning were introduced into deep learning and implemented in the InfiniBand network.

Federated Learning: SMC & Homomorphic Encryption

- Already discussed earlier....

Federated Learning: Categories: HFL

Based on the characteristics of data distribution, FL can be generally categorized into three classes as follows:

- Horizontal FL OR sample-based: used in those cases, in which datasets of multiple clients share the same feature space but different space in samples, to construct a unified model across different clients.

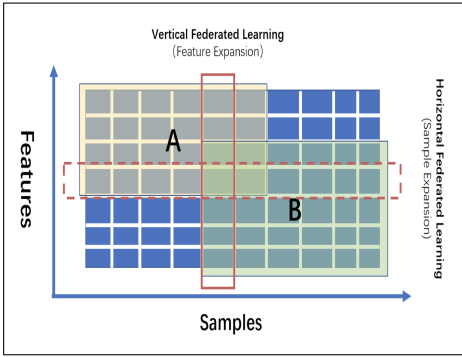

Figure: Horizontal/Vertical FL

Federated Learning: Categories: HFL

Based on the characteristics of data distribution, FL can be generally categorized into three classes as follows:

- Horizontal FL OR sample-based: used in those cases, in which datasets of multiple clients share the same feature space but different space in samples, to construct a unified model across different clients.
- Formally, supposing that D represents data, X represents samples, Y represents features, and I represents the data index.

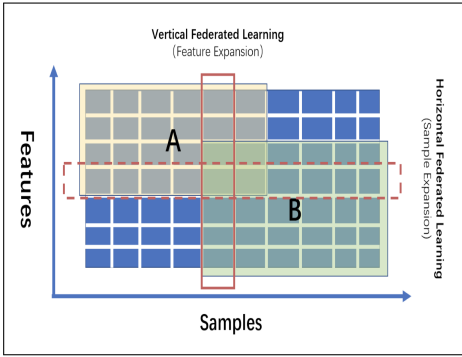

Figure: Horizontal/Vertical FL

Federated Learning: Categories: HFL

Based on the characteristics of data distribution, FL can be generally categorized into three classes as follows:

- Horizontal FL OR sample-based: used in those cases, in which datasets of multiple clients share the same feature space but different space in samples, to construct a unified model across different clients.
- Formally, supposing that D represents data, X represents samples, Y represents features, and I represents the data index.
  - Then, horizontal federal learning can be represented as:

$$X_i \neq X_j, Y_i = Y_j, I_i \neq I_j, \forall\, D_i, D_j, i \neq j \tag{1}$$

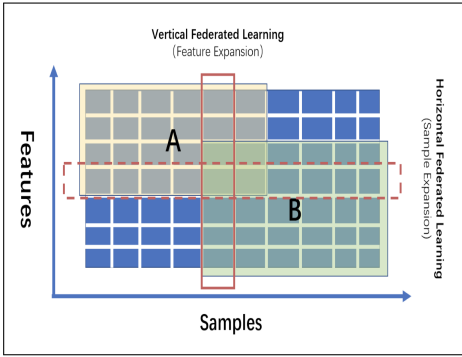

Figure: Horizontal/Vertical FL

Federated Learning: Categories: HFL

Based on the characteristics of data distribution, FL can be generally categorized into three classes as follows:

- Horizontal FL OR sample-based: used in those cases, in which datasets of multiple clients share the same feature space but different space in samples, to construct a unified model across different clients.
- Formally, supposing that D represents data, X represents samples, Y represents features, and I represents the data index.
  - Then, horizontal federal learning can be represented as:

$$X_i \neq X_j, Y_i = Y_j, I_i \neq I_j, \forall\, D_i, D_j, i \neq j \tag{1}$$

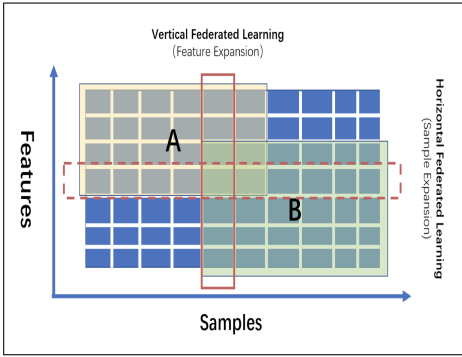

Figure: Horizontal/Vertical FL

Federated Learning: Categories: HFL...

- Horizontal FL OR sample-based: the data features X_1,X_2,..... (and some label data Y) of data sets of two data holders overlap largely, while the data samples of users U_1,U_2,. . . minimally overlap.

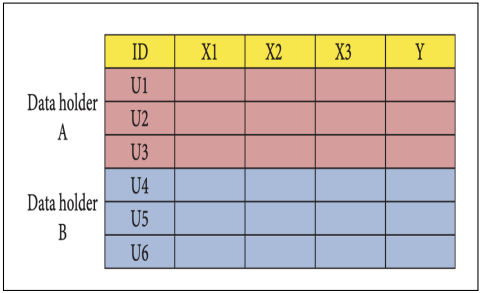

Figure: Horizontal FL

1

[1] Src: Zhang, Xiaodong & Guo, Chunrong. (2022). Research on Open Innovation Intelligent Decision-Making of Cross-Border E-Commerce Based on Federated Learning. Mathematical Problems in Engineering. 2022. 1-13.

Federated Learning: Categories: HFL...

- Horizontal FL OR sample-based: the data features X_1,X_2,..... (and some label data Y) of data sets of two data holders overlap largely, while the data samples of users U_1,U_2,. . . minimally overlap.
- Competitors of the same type of business (and hence same features) but distributed in different regions have different user groups or come from different regions

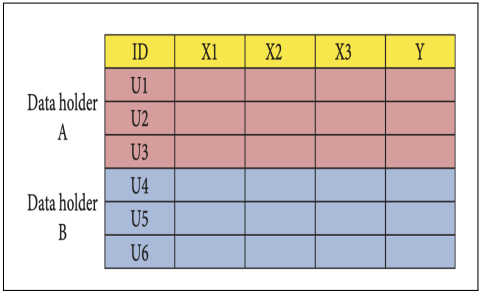

Figure: Horizontal FL

1

[1] Src: Zhang, Xiaodong & Guo, Chunrong. (2022). Research on Open Innovation Intelligent Decision-Making of Cross-Border E-Commerce Based on Federated Learning. Mathematical Problems in Engineering. 2022. 1-13.

Federated Learning: Categories: HFL...

- Horizontal FL OR sample-based: the data features X_1,X_2,..... (and some label data Y) of data sets of two data holders overlap largely, while the data samples of users U_1,U_2,. . . minimally overlap.
- Competitors of the same type of business (and hence same features) but distributed in different regions have different user groups or come from different regions
  - hence, their intersection or overlap is very small.

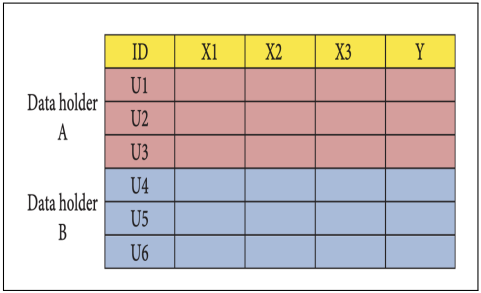

Figure: Horizontal FL

1

[1] Src: Zhang, Xiaodong & Guo, Chunrong. (2022). Research on Open Innovation Intelligent Decision-Making of Cross-Border E-Commerce Based on Federated Learning. Mathematical Problems in Engineering. 2022. 1-13.

Federated Learning: Categories: HFL...

- Horizontal FL OR sample-based: the data features X_1,X_2,..... (and some label data Y) of data sets of two data holders overlap largely, while the data samples of users U_1,U_2,. . . minimally overlap.
- Competitors of the same type of business (and hence same features) but distributed in different regions have different user groups or come from different regions
  - hence, their intersection or overlap is very small.
  - thus, the joint training of this type of enterprises is done using HFL.

[1]

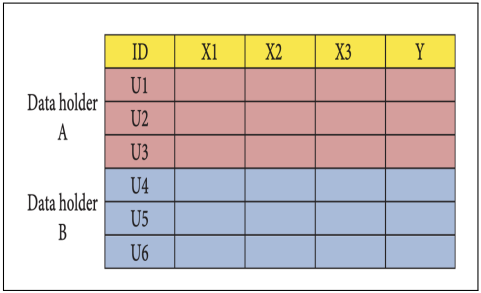

Figure: Horizontal FL

[1] Src: Zhang, Xiaodong & Guo, Chunrong. (2022). Research on Open Innovation Intelligent Decision-Making of Cross-Border E-Commerce Based on Federated Learning. Mathematical Problems in Engineering. 2022. 1-13.

Federated Learning: Categories: HFL...

Thus, the main idea of HFL is

- to help multiple users using their own data to jointly train a reliable model, while ensuring the privacy and security of data.

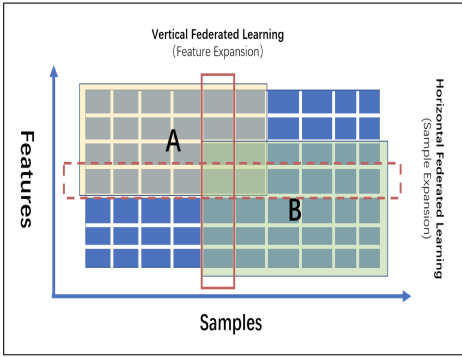

Figure: Horizontal/Vertical FL

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Federated Learning: Categories: HFL...

Thus, the main idea of HFL is

- to help multiple users using their own data to jointly train a reliable model, while ensuring the privacy and security of data.
- However, for sample expansions, the data of all parties need to be aligned first to ensure that all parties involved in the training have the same feature domain - to build the same model architecture and iterate synchronously.

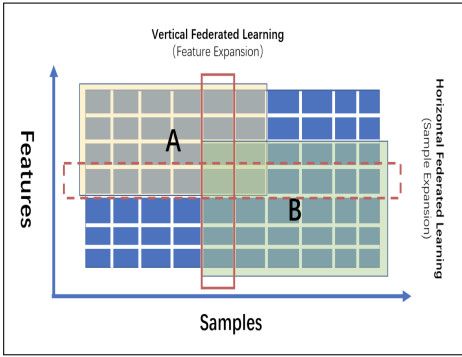

Figure: Horizontal/Vertical FL

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Federated Learning: Categories: HFL...

Thus, the main idea of HFL is

- to help multiple users using their own data to jointly train a reliable model, while ensuring the privacy and security of data.
- However, for sample expansions, the data of all parties need to be aligned first to ensure that all parties involved in the training have the same feature domain - to build the same model architecture and iterate synchronously.
- Google makes use of HFL in the Gboard mobile keyboard for out-of-vocabulary word prediction and emoji prediction,

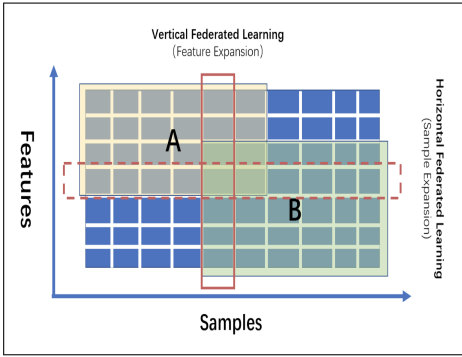

Figure: Horizontal/Vertical FL

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Federated Learning: Categories: HFL...

Thus, the main idea of HFL is

- to help multiple users using their own data to jointly train a reliable model, while ensuring the privacy and security of data.
- However, for sample expansions, the data of all parties need to be aligned first to ensure that all parties involved in the training have the same feature domain - to build the same model architecture and iterate synchronously.
- Google makes use of HFL in the Gboard mobile keyboard for out-of-vocabulary word prediction and emoji prediction,
- Used in Android messages also for providing personalized suggestions.

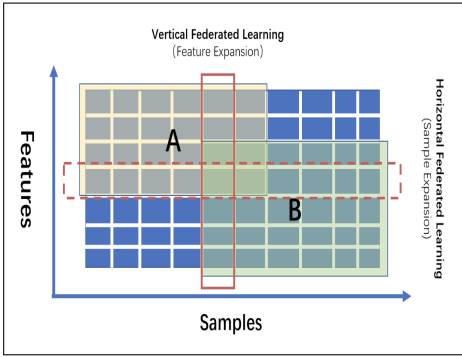

Figure: Horizontal/Vertical FL

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Federated Learning: Categories: Vertical FL

Vertical FL OR feature-based FL

- is applicable to the scenarios in which the datasets come from different domains,

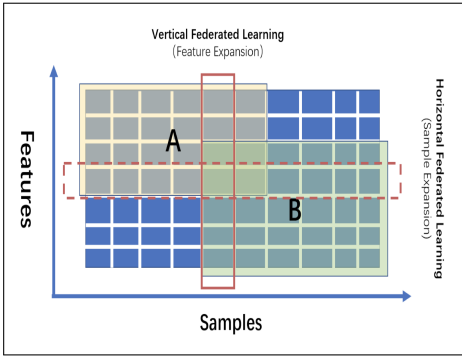

Figure: Horizontal/Vertical FL

1

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Federated Learning: Categories: Vertical FL

Vertical FL OR feature-based FL

- is applicable to the scenarios in which the datasets come from different domains,
- share the same sample space but differ in feature space,

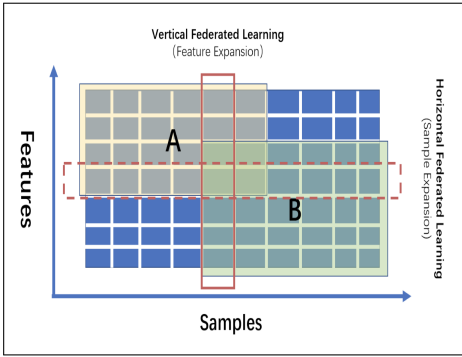

Figure: Horizontal/Vertical FL

1

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Federated Learning: Categories: Vertical FL

Vertical FL OR feature-based FL

- is applicable to the scenarios in which the datasets come from different domains,
- share the same sample space but differ in feature space,
- with a goal to construct a unified model across different clients.

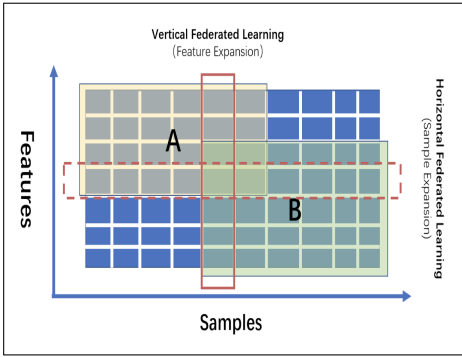

Figure: Horizontal/Vertical FL

1

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Federated Learning: Categories: VFL...

- Vertical FL OR feature-based: here, the user samples U_1,U_2,. . . in the data sets of the data holder have a large overlap, whereas the data features X_1,X_2,..... minimally overlap.

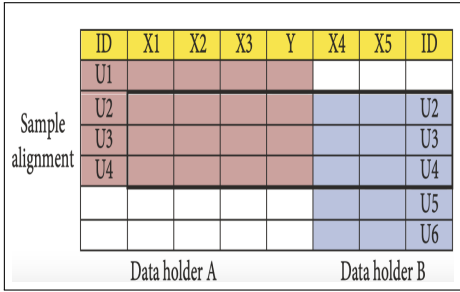

Figure: Vertical FL

[1] [1] Src: Zhang, Xiaodong & Guo, Chunrong. (2022). Research on Open Innovation Intelligent Decision-Making of Cross-Border E-Commerce Based on Federated Learning. Mathematical Problems in Engineering. 2022. 1-13.

Federated Learning: Categories: VFL...

- Vertical FL OR feature-based: here, the user samples U_1,U_2,. . . in the data sets of the data holder have a large overlap, whereas the data features X_1,X_2,..... minimally overlap.
- then, the portion of data with the same users but different data features is taken out for training

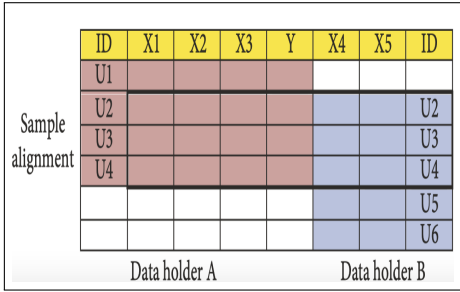

Figure: Vertical FL

[1] [1] Src: Zhang, Xiaodong & Guo, Chunrong. (2022). Research on Open Innovation Intelligent Decision-Making of Cross-Border E-Commerce Based on Federated Learning. Mathematical Problems in Engineering. 2022. 1-13.

Federated Learning: Categories: VFL...

- Vertical FL OR feature-based: here, the user samples U_1,U_2,. . . in the data sets of the data holder have a large overlap, whereas the data features X_1,X_2,..... minimally overlap.
- then, the portion of data with the same users but different data features is taken out for training
- For example, banks and e-commerce enterprises.

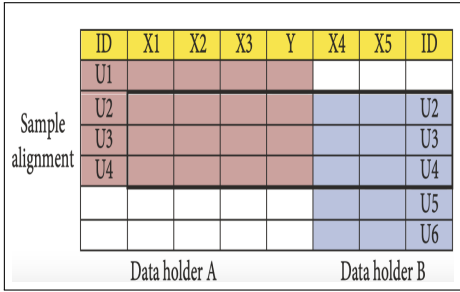

Figure: Vertical FL

[1] [1] Src: Zhang, Xiaodong & Guo, Chunrong. (2022). Research on Open Innovation Intelligent Decision-Making of Cross-Border E-Commerce Based on Federated Learning. Mathematical Problems in Engineering. 2022. 1-13.

Federated Learning: Categories: VFL...

- Vertical FL OR feature-based: here, the user samples U_1,U_2,. . . in the data sets of the data holder have a large overlap, whereas the data features X_1,X_2,..... minimally overlap.
- then, the portion of data with the same users but different data features is taken out for training
- For example, banks and e-commerce enterprises.
  - Banks may record users' income, expenditure and credit rating, whereas e-commerce enterprises may record users' browsing and purchasing information.

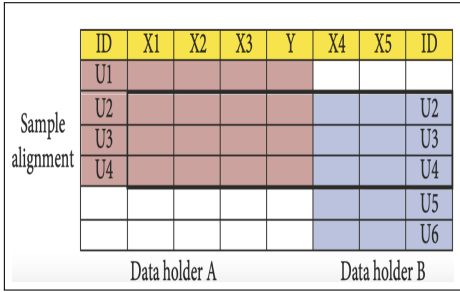

Figure: Vertical FL

1 [1] Src: Zhang, Xiaodong & Guo, Chunrong. (2022). Research on Open Innovation Intelligent Decision-Making of Cross-Border E-Commerce Based on Federated Learning. Mathematical Problems in Engineering. 2022. 1-13.

Federated Learning: Categories: VFL...

- Vertical FL OR feature-based: here, the user samples $U_1, U_2, \ldots$ in the data sets of the data holder have a large overlap, whereas the data features $X_1, X_2, \ldots$ minimally overlap.
- then, the portion of data with the same users but different data features is taken out for training
- For example, banks and e-commerce enterprises.
  - Banks may record users' income, expenditure and credit rating, whereas e-commerce enterprises may record users' browsing and purchasing information.
  - thus, the intersection of their data features is small, but the intersection/overlap of their users is large.

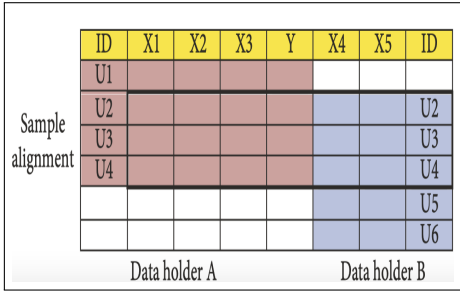

Figure: Vertical FL

[1] [1] Src: Zhang, Xiaodong & Guo, Chunrong. (2022). Research on Open Innovation Intelligent Decision-Making of Cross-Border E-Commerce Based on Federated Learning. Mathematical Problems in Engineering. 2022. 1-13.

Federated Learning: Categories...

Federated transfer learning

- is used where the datasets differ not only in samples but also in feature space, to construct a unified model across different clients.

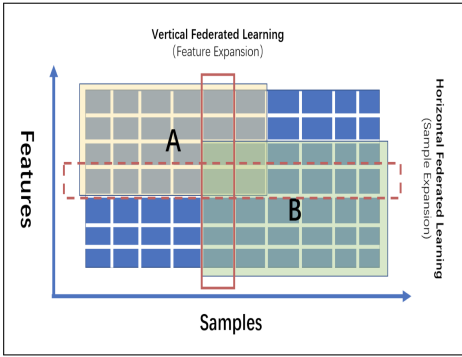

Figure: Horizontal/Vertical FL

1

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Federated Learning: Architecture: Generic view

Figure here shows a general architecture of a Federated Neural Network

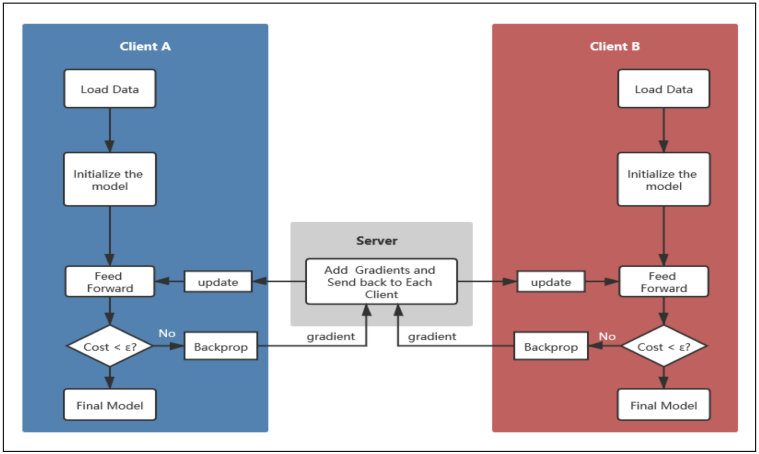

Figure: General architecture of a Federated Neural Network

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Federated Learning: Architecture...

The architecture shown in the figure early is motivated by the following:

- to help all parties jointly train the same model by passing intermediate variables in the training process.

Federated Learning: Architecture...

The architecture shown in the figure early is motivated by the following:

- to help all parties jointly train the same model by passing intermediate variables in the training process.
- Since, most neural networks are trained by gradient descent, in this architecture, gradients are chosen as its intermediate variables.

Federated Learning: Architecture...

The architecture shown in the figure early is motivated by the following:

- to help all parties jointly train the same model by passing intermediate variables in the training process.
- Since, most neural networks are trained by gradient descent, in this architecture, gradients are chosen as its intermediate variables.
- Although the gradient cannot represent all the data directly, it can represent the relationship between the model and the data which facilitate model training.

Federated Learning: Architecture...

The architecture shown in the figure early is motivated by the following:

- to help all parties jointly train the same model by passing intermediate variables in the training process.
- Since, most neural networks are trained by gradient descent, in this architecture, gradients are chosen as its intermediate variables.
- Although the gradient cannot represent all the data directly, it can represent the relationship between the model and the data which facilitate model training.
- As shown the architecture consists of learning clients and a computing server.

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.
- their main functions include

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.
- their main functions include
  - initializing the same initial model with other clients,

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.
- their main functions include
  - initializing the same initial model with other clients,
  - training data locally,

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.
- their main functions include
  - initializing the same initial model with other clients,
  - training data locally,
  - extracting gradients during the training,

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.
- their main functions include
  - initializing the same initial model with other clients,
  - training data locally,
  - extracting gradients during the training,
  - computing the gradients with computing server,

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.
- their main functions include
  - initializing the same initial model with other clients,
  - training data locally,
  - extracting gradients during the training,
  - computing the gradients with computing server,
  - collecting server responses, passing the results & updating the model, and

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.
- their main functions include
  - initializing the same initial model with other clients,
  - training data locally,
  - extracting gradients during the training,
  - computing the gradients with computing server,
  - collecting server responses, passing the results & updating the model, and
  - iterating repeatedly until the model converges.

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.
- their main functions include
  - initializing the same initial model with other clients,
  - training data locally,
  - extracting gradients during the training,
  - computing the gradients with computing server,
  - collecting server responses, passing the results & updating the model, and
  - iterating repeatedly until the model converges.
- The computing server is an intermediate platform in the learning process and its main functions are

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.
- their main functions include
  - initializing the same initial model with other clients,
  - training data locally,
  - extracting gradients during the training,
  - computing the gradients with computing server,
  - collecting server responses, passing the results & updating the model, and
  - iterating repeatedly until the model converges.
- The computing server is an intermediate platform in the learning process and its main functions are
  - receiving the gradient information from multiple learning clients,

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.
- their main functions include
  - initializing the same initial model with other clients,
  - training data locally,
  - extracting gradients during the training,
  - computing the gradients with computing server,
  - collecting server responses, passing the results & updating the model, and
  - iterating repeatedly until the model converges.
- The computing server is an intermediate platform in the learning process and its main functions are
  - receiving the gradient information from multiple learning clients,
  - performing calculations on the gradients,

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.
- their main functions include
  - initializing the same initial model with other clients,
  - training data locally,
  - extracting gradients during the training,
  - computing the gradients with computing server,
  - collecting server responses, passing the results & updating the model, and
  - iterating repeatedly until the model converges.
- The computing server is an intermediate platform in the learning process and its main functions are
  - receiving the gradient information from multiple learning clients,
  - performing calculations on the gradients,
  - integrating the information learned by multiple models, and

Federated Learning: Architecture...

As shown the architecture consists of learning clients and a computing server. The learning clients,

- have their own private data and, supposing all the data have been aligned, their quantitative dimensions with other learning participants.
- their main functions include
  - initializing the same initial model with other clients,
  - training data locally,
  - extracting gradients during the training,
  - computing the gradients with computing server,
  - collecting server responses, passing the results & updating the model, and
  - iterating repeatedly until the model converges.
- The computing server is an intermediate platform in the learning process and its main functions are
  - receiving the gradient information from multiple learning clients,
  - performing calculations on the gradients,
  - integrating the information learned by multiple models, and
  - transmitting the result to each learning client separately.

Federated learning process: Schematic Diagram

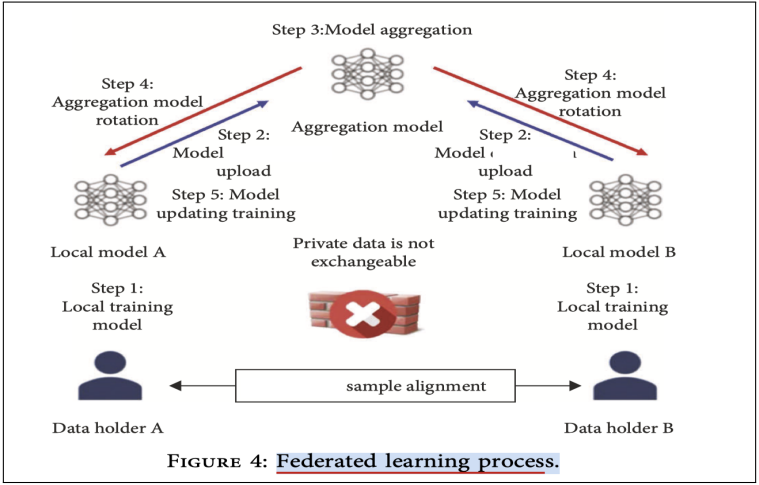

FIGURE 4: Federated learning process.

Figure: Federated learning process

[1] Src: Zhang, Xiaodong & Guo, Chunrong. (2022). Research on Open Innovation Intelligent Decision-Making of Cross-Border E-Commerce Based on Federated Learning. Mathematical Problems in Engineering. 2022. 1-13.

Federated Learning: Federated Network Algorithm

Assuming that

- there are K federated clients with the same data structure and feature space

Federated Learning: Federated Network Algorithm

Assuming that

- there are K federated clients with the same data structure and feature space
- the K federated clients collaboratively train a unified deep learning model with the coordination of a central server.

Federated Learning: Federated Network Algorithm

Assuming that

- there are K federated clients with the same data structure and feature space
- the K federated clients collaboratively train a unified deep learning model with the coordination of a central server.
- the central server is responsible for the model training process, and repeats the steps shown below, until the training is stopped,

Federated Learning: Federated Network Algorithm

Assuming that

- there are K federated clients with the same data structure and feature space
- the K federated clients collaboratively train a unified deep learning model with the coordination of a central server.
- the central server is responsible for the model training process, and repeats the steps shown below, until the training is stopped,
- this is such that at the i^{th} training round ($i \epsilon E$, E = the number of training rounds), the updating process of the global model M is performed as follows:

Federated Learning: Federated Network Algorithm...

This is such that at the i^{th} training round ($i \in E$, E = the number of training rounds), the updating process of the global model M is performed as follows:

1. The K federated clients download the current global model M^i and a training setting from the central server.
2. Each federated client C_k(k $\in$ { 1, 2,.....,K}) trains the downloaded model M^i on its local data D_k for E_{local} rounds based on the training setting, and then computes an update U_i^k with respect to M^i
3. The central server collects all the updates $U_i = \{ U_i^1, U_i^2, ..., U_i^K \}$ from the K federated clients.
4. The central server updates the global model based on the aggregation of the collected updates U_i, thereby obtaining an updated model M^{i+1} that will play the role as the global model for the next training round.
5. When a termination criterion has been satisfied, the central server will stop the above iterative training process and get the final FL model.

Federated Multi-Layer Perceptron Algorithm (FMLP)

The multilayer perceptron (MLP)(s),

- is an algorithm for predicting a categorical (classification) or continuous (regression) target variable.

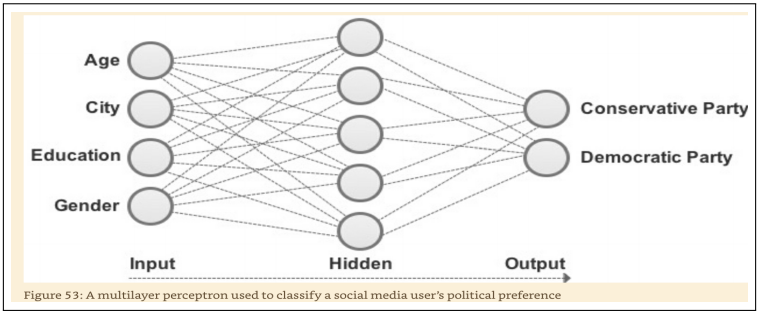

Figure: General architecture of a Multilayer Perceptron i.e. ML Neural Network

[1] Src: Theobald, Oliver. Machine Learning for Absolute Beginners: A Plain English Introduction (3rd Edition) (Machine Learning with Python for Beginners Book 1) (p. 109)

The multilayer perceptron (MLP)(s),

- is an algorithm for predicting a categorical (classification) or continuous (regression) target variable.
- are powerful because they aggregate multiple models into a unified prediction model, as demonstrated by the classification model shown here.

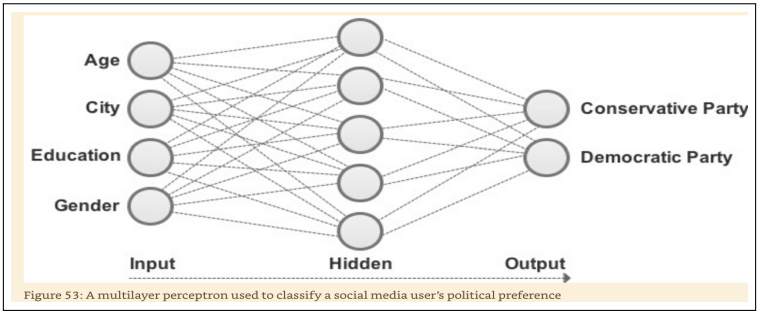

Figure 53: A multilayer perceptron used to classify a social media user's political preference

Figure: General architecture of a Multilayer Perceptron i.e. ML Neural Network

[1] Src: Theobald, Oliver. Machine Learning for Absolute Beginners: A Plain English Introduction (3rd Edition) (Machine Learning with Python for Beginners Book 1) (p. 109)

The multilayer perceptron (MLP)(s),

- is an algorithm for predicting a categorical (classification) or continuous (regression) target variable.
- are powerful because they aggregate multiple models into a unified prediction model, as demonstrated by the classification model shown here.

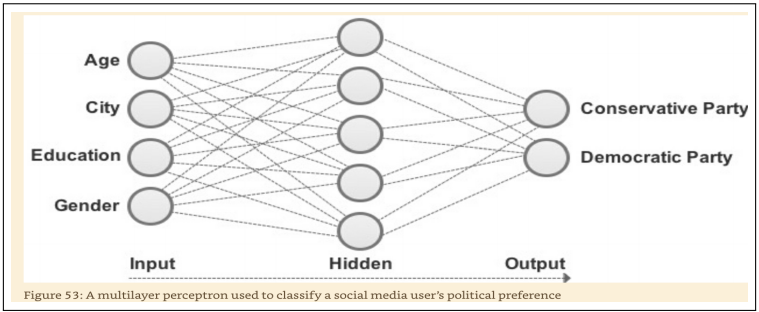

Figure 53: A multilayer perceptron used to classify a social media user's political preference

Figure: General architecture of a Multilayer Perceptron i.e. ML Neural Network

[1] Src: Theobald, Oliver. Machine Learning for Absolute Beginners: A Plain English Introduction (3rd Edition) (Machine Learning with Python for Beginners Book 1) (p. 109)

The multilayer perceptron (MLP)(s),

- is an algorithm for predicting a categorical (classification) or continuous (regression) target variable.
- are powerful because they aggregate multiple models into a unified prediction model, as demonstrated by the classification model shown here.

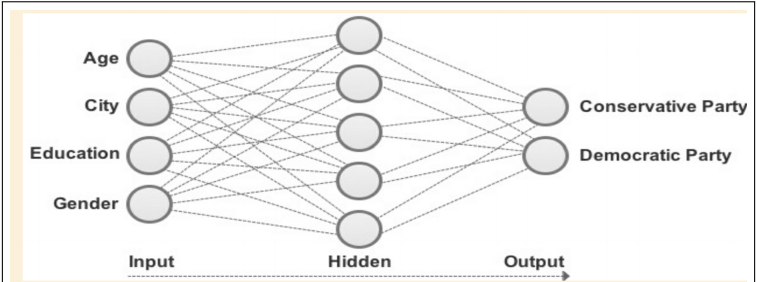

Figure 53: A multilayer perceptron used to classify a social media user's political preference

1 Figure: General architecture of a Multilayer Perceptron i.e. ML Neural Network

[1] Src: Theobald, Oliver. Machine Learning for Absolute Beginners: A Plain English Introduction (3rd Edition) (Machine Learning with Python for Beginners Book 1) (p. 109)

Multi-Layer Perceptron ... Src: wiki

A multilayer perceptron (MLP)(s)

- consists of at least three layers of nodes: an input layer, a hidden layer and an output layer.

Multi-Layer Perceptron ... Src: wiki

A multilayer perceptron (MLP)(s)

- consists of at least three layers of nodes: an input layer, a hidden layer and an output layer.
- except for the input nodes, each node is a neuron that uses a nonlinear activation function.

Multi-Layer Perceptron ... Src: wiki

A multilayer perceptron (MLP)(s)

- consists of at least three layers of nodes: an input layer, a hidden layer and an output layer.
- except for the input nodes, each node is a neuron that uses a nonlinear activation function.
- a supervised learning technique called backpropagation algorithm for training.

Multi-Layer Perceptron ... Src: wiki

A multilayer perceptron (MLP)(s)

- consists of at least three layers of nodes: an input layer, a hidden layer and an output layer.
- except for the input nodes, each node is a neuron that uses a nonlinear activation function.
- a supervised learning technique called backpropagation algorithm for training.
- the multiple layers and non-linear activation distinguish MLP from a linear perceptron.

Multi-Layer Perceptron ... Src: wiki

A multilayer perceptron (MLP)(s)

- consists of at least three layers of nodes: an input layer, a hidden layer and an output layer.
- except for the input nodes, each node is a neuron that uses a nonlinear activation function.
- a supervised learning technique called backpropagation algorithm for training.
- the multiple layers and non-linear activation distinguish MLP from a linear perceptron.

Multi-Layer Perceptron ...

Src: wiki

A multilayer perceptron (MLP)(s)

- consists of at least three layers of nodes: an input layer, a hidden layer and an output layer.
- except for the input nodes, each node is a neuron that uses a nonlinear activation function.
- a supervised learning technique called backpropagation algorithm for training.
- the multiple layers and non-linear activation distinguish MLP from a linear perceptron.

Obviously, if a multilayer perceptron has a linear activation function in all neurons,

- that is, a linear function that maps the weighted inputs to the output of each neuron, then linear algebra shows that any number of layers can be reduced to a two-layer input-output model.

Multi-Layer Perceptron ... Src: wiki

A multilayer perceptron (MLP)(s)

- consists of at least three layers of nodes: an input layer, a hidden layer and an output layer.
- except for the input nodes, each node is a neuron that uses a nonlinear activation function.
- a supervised learning technique called backpropagation algorithm for training.
- the multiple layers and non-linear activation distinguish MLP from a linear perceptron.

Obviously, if a multilayer perceptron has a linear activation function in all neurons,

- that is, a linear function that maps the weighted inputs to the output of each neuron, then linear algebra shows that any number of layers can be reduced to a two-layer input-output model.
- but, some neurons in MLP use a nonlinear activation function - developed to model the frequency of action potentials, or firing, of biological neurons.

Federated Multi-Layer Perceptron Algorithm

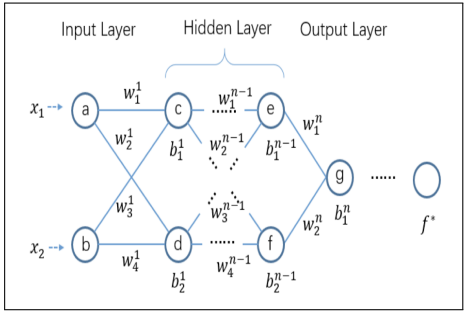

Figure: Another architecture

#	Parameter	Meaning
1	x	the sample in the dataset
2	θ	the parameters of the model
3	f_p	feed forward process
4	*out*	the output of each iteration
5	$f*$	activation function[1]
6	*loss*	loss function
7	c	loss calculated by loss function
8	ϵ	minimum error
9	b_p	back-propagation process
10	*grad*	gradient calculated by bp process
11	l_r	learning rate

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Federated Multi-Layer Perceptron Algorithm

- The parameter of the model is $\theta = \{\omega_1 ..... \ \omega_n, \ b_1 ..... \ b_n\}$, the learning rate of training is l_r, the data set represented as $x = \{x_1 .....x_n\}$.

.....*continued*

Federated Multi-Layer Perceptron Algorithm

- The parameter of the model is $\theta = \{\omega_1 ..... \ \omega_n, \ b_1 ..... \ b_n\}$, the learning rate of training is l_r, the data set represented as $x = \{x_1 .....x_n\}$.
- The purpose of the model is to approximate a distribution f^*.

.....continued

Federated Multi-Layer Perceptron Algorithm

- The parameter of the model is $\theta = \{\omega_1 ..... \ \omega_n, \ b_1 ..... \ b_n\}$, the learning rate of training is l_r, the data set represented as $x = \{x_1 .....x_n\}$.
- The purpose of the model is to approximate a distribution f^*.
- The forward process of the network is to calculate the output of the training defined as: $out = fp(x, \theta)$

.....*continued*

Federated Multi-Layer Perceptron Algorithm

- The parameter of the model is $\theta = \{\omega_1 ..... \ \omega_n, \ b_1 ..... \ b_n\}$, the learning rate of training is l_r, the data set represented as $x = \{x_1 .....x_n\}$.
- The purpose of the model is to approximate a distribution f^*.
- The forward process of the network is to calculate the output of the training defined as: $out = fp(x, \theta)$
- The loss function that calculates the distance between the output and the ideal value is $c = loss(f^*(x), \ out)$

.....*continued*

Federated Multi-Layer Perceptron Algorithm

- The parameter of the model is $\theta = \{\omega_1 ..... \ \omega_n, \ b_1 ..... \ b_n\}$, the learning rate of training is l_r, the data set represented as $x = \{x_1 .....x_n\}$.
- The purpose of the model is to approximate a distribution f^*.
- The forward process of the network is to calculate the output of the training defined as: $out = fp(x, \theta)$
- The loss function that calculates the distance between the output and the ideal value is $c = loss(f^*(x), \ out)$
- The function of the back-propagation is

.....*continued*

Federated Multi-Layer Perceptron Algorithm

- The parameter of the model is $\theta = \{\omega_1 ..... \omega_n, b_1 ..... b_n\}$, the learning rate of training is l_r, the data set represented as $x = \{x_1 .....x_n\}$.
- The purpose of the model is to approximate a distribution f^*.
- The forward process of the network is to calculate the output of the training defined as: $out = fp(x, \theta)$
- The loss function that calculates the distance between the output and the ideal value is $c = loss(f^*(x), out)$
- The function of the back-propagation is
  - to calculate the gradients and to propagate them from the loss function backwards

.....*continued*

Federated Multi-Layer Perceptron Algorithm

- The parameter of the model is $\theta = \{\omega_1 ..... \omega_n, b_1 ..... b_n\}$, the learning rate of training is l_r, the data set represented as $x = \{x_1 .....x_n\}$.
- The purpose of the model is to approximate a distribution f^*.
- The forward process of the network is to calculate the output of the training defined as: $out = fp(x, \theta)$
- The loss function that calculates the distance between the output and the ideal value is $c = loss(f^*(x), out)$
- The function of the back-propagation is
  - to calculate the gradients and to propagate them from the loss function backwards
  - this is to help the network adjust the parameters according to the gradient to reduce the error between the output value and the ideal one.

.....*continued*

Federated Multi-Layer Perceptron Algorithm

- The parameter of the model is $\theta = \{\omega_1 ..... \omega_n, b_1 ..... b_n\}$, the learning rate of training is l_r, the data set represented as $x = \{x_1 .....x_n\}$.
- The purpose of the model is to approximate a distribution f^*.
- The forward process of the network is to calculate the output of the training defined as: $out = fp(x, \theta)$
- The loss function that calculates the distance between the output and the ideal value is $c = loss(f^*(x), out)$
- The function of the back-propagation is
  - to calculate the gradients and to propagate them from the loss function backwards
  - this is to help the network adjust the parameters according to the gradient to reduce the error between the output value and the ideal one.
- What about the back-propagation process?

.....*continued*

Federated Multi-Layer Perceptron Algorithm

- The parameter of the model is $\theta = \{\omega_1 ..... \omega_n, b_1 ..... b_n\}$, the learning rate of training is l_r, the data set represented as $x = \{x_1 .....x_n\}$.
- The purpose of the model is to approximate a distribution f^*.
- The forward process of the network is to calculate the output of the training defined as: $out = fp(x, \theta)$
- The loss function that calculates the distance between the output and the ideal value is $c = loss(f^*(x),\ out)$
- The function of the back-propagation is
  - to calculate the gradients and to propagate them from the loss function backwards
  - this is to help the network adjust the parameters according to the gradient to reduce the error between the output value and the ideal one.
- What about the back-propagation process?
- The back-propagation process can be defined as $grad = bp(x,\ \theta,\ c)$

.....*continued*

Federated Multi-Layer Perceptron Algorithm...

What about the model-update process?

- The model-update process is to adjust the network parameters based on the gradient obtained by backpropagation, which is expressed as: $\theta' = \theta$ - $l_r \cdot grad$

Federated Multi-Layer Perceptron Algorithm...

What about the model-update process?

- The model-update process is to adjust the network parameters based on the gradient obtained by backpropagation, which is expressed as: $\theta' = \theta$ - $l_r \cdot grad$
- Through the federated network realized by MLP, we can get a federated MLP (FMLP).

Federated Multi-Layer Perceptron Algorithm...

What about the model-update process?

- The model-update process is to adjust the network parameters based on the gradient obtained by backpropagation, which is expressed as: $\theta' = \theta - l_r \cdot grad$
- Through the federated network realized by MLP, we can get a federated MLP (FMLP).
- Then, a copy of the MLP model is stored in the local memory of each learning client.

Federated Multi-Layer Perceptron Algorithm...

What about the model-update process?

- The model-update process is to adjust the network parameters based on the gradient obtained by backpropagation, which is expressed as: $\theta' = \theta$ - $l_r \cdot grad$
- Through the federated network realized by MLP, we can get a federated MLP (FMLP).
- Then, a copy of the MLP model is stored in the local memory of each learning client.
- It contains an input layer with x units, n hidden layers with y units each, and an output layer with z units.

Federated Multi-Layer Perceptron Algorithm...

What about the model-update process?

- The model-update process is to adjust the network parameters based on the gradient obtained by backpropagation, which is expressed as: $\theta' = \theta - l_r \cdot grad$
- Through the federated network realized by MLP, we can get a federated MLP (FMLP).
- Then, a copy of the MLP model is stored in the local memory of each learning client.
- It contains an input layer with x units, n hidden layers with y units each, and an output layer with z units.
- The size of x depends on the feature dimensions of the input data.

Federated Multi-Layer Perceptron Algorithm...

What about the model-update process?

- The model-update process is to adjust the network parameters based on the gradient obtained by backpropagation, which is expressed as: $\theta' = \theta - l_r \cdot grad$
- Through the federated network realized by MLP, we can get a federated MLP (FMLP).
- Then, a copy of the MLP model is stored in the local memory of each learning client.
- It contains an input layer with x units, n hidden layers with y units each, and an output layer with z units.
- The size of x depends on the feature dimensions of the input data.
- In addition, the size of z depends on the required output of the network that is closely dependent on target output of the real applications.

Federated Multi-Layer Perceptron Algorithm...

What about the model-update process?

- The model-update process is to adjust the network parameters based on the gradient obtained by backpropagation, which is expressed as: $\theta' = \theta - l_r \cdot grad$
- Through the federated network realized by MLP, we can get a federated MLP (FMLP).
- Then, a copy of the MLP model is stored in the local memory of each learning client.
- It contains an input layer with x units, n hidden layers with y units each, and an output layer with z units.
- The size of x depends on the feature dimensions of the input data.
- In addition, the size of z depends on the required output of the network that is closely dependent on target output of the real applications.

Federated Multi-Layer Perceptron Algorithm...

What about the model-update process?

- The model-update process is to adjust the network parameters based on the gradient obtained by backpropagation, which is expressed as: $\theta' = \theta - l_r \cdot grad$
- Through the federated network realized by MLP, we can get a federated MLP (FMLP).
- Then, a copy of the MLP model is stored in the local memory of each learning client.
- It contains an input layer with x units, n hidden layers with y units each, and an output layer with z units.
- The size of x depends on the feature dimensions of the input data.
- In addition, the size of z depends on the required output of the network that is closely dependent on target output of the real applications.

Then the pseudocode of the algorithm is as shown on the next slide.....

Federated Multi-Layer Perceptron Algorithm...

Algorithm 1 Federated Multi-Layer Perceptron

Input: Dataset x
Output: Model θ_{final}

1: Initialize model parameters θ
2: **for** i in *iteration* **do**
3: Forward propagation: $out_i = fp(x_i, \theta_i)$;
4: Compute loss: $c_i = loss(f^*(x_i), out_i)$;
5: **if** $c_i < \epsilon$ **then**
6: Break
7: **else**
8: Back propagation: $grad_i = bp(x_i, \theta_i, c_i)$;
9: Send gradients to computing server and get new gradients;
10: Update: $\theta_{i+1} = \theta_i - lr * grad_{new}$;
11: **end if**
12: **end for**
13: **return** Model with parameters θ_{final}

#	Parameter	Meaning
1	x	the sample in the dataset
2	θ	the parameters of the model
3	f_p	feed forward process
4	*out*	the output of each iteration
5	$f*$	activation function[1]
6	*loss*	loss function
7	c	loss calculated by loss function
8	ϵ	minimum error
9	b_p	back-propagation process
10	*grad*	gradient calculated by bp process
11	l_r	learning rate

1

Figure: Federated Multi-Layer Perceptron Algorithm

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Privacy Preserving ML with Partially Homomorphic Encryption & Federated Learning

What are the issues with this model ?

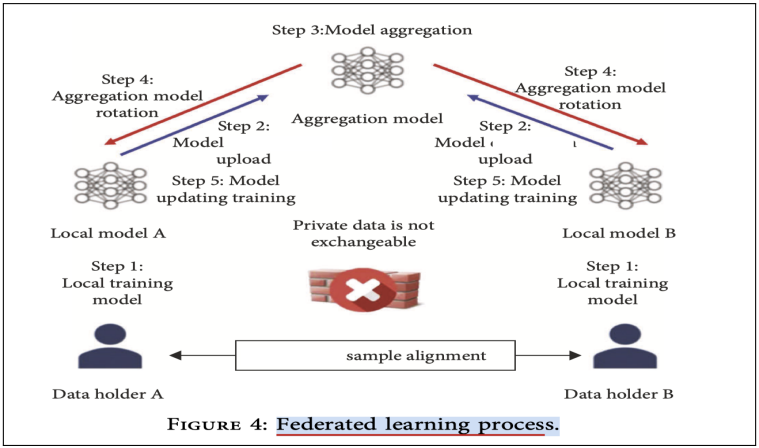

FIGURE 4: Federated learning process.

Figure: Federated learning process

[1] Src: Zhang, Xiaodong & Guo, Chunrong. (2022). Research on Open Innovation Intelligent Decision-Making of Cross-Border E-Commerce Based on Federated Learning. Mathematical Problems in Engineering. 2022. 1-13.

What are the issues with this model ? What is then required ? How can it be achieved?

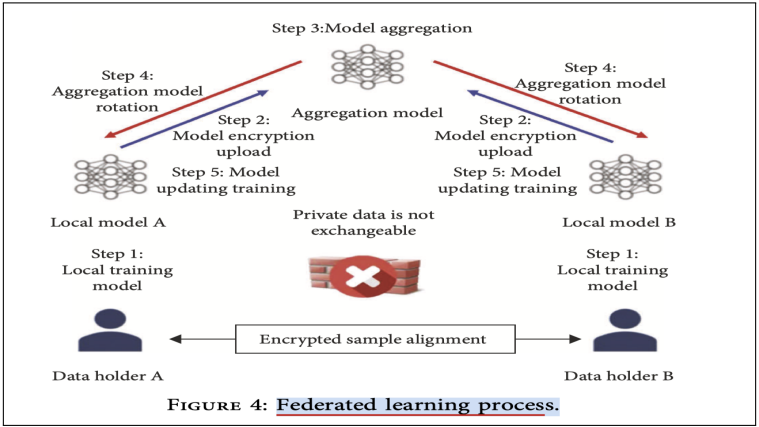

FIGURE 4: Federated learning process.

Figure: Federated learning process, improved

[1] Src: Zhang, Xiaodong & Guo, Chunrong. (2022). Research on Open Innovation Intelligent Decision-Making of Cross-Border E-Commerce Based on Federated Learning. Mathematical Problems in Engineering. 2022. 1-13.

- Study of the research efforts published in [1].

[1] Fang, H.; Qian, Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet 2021, 13, 94

- Study of the research efforts published in [1].
- The authors propose, a multi-party PPML framework, that is based on

[1] Fang, H.; Qian, Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet 2021, 13, 94

- Study of the research efforts published in [1].
- The authors propose, a multi-party PPML framework, that is based on
  - partially homomorphic encryption using an improved Paillier algorithm to speed up the training by 25–28%

[1] Fang, H.; Qian, Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet 2021, 13, 94

- Study of the research efforts published in [1].
- The authors propose, a multi-party PPML framework, that is based on
  - partially homomorphic encryption using an improved Paillier algorithm to speed up the training by 25–28%
  - federated learning

[1] Fang, H.; Qian, Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet 2021, 13, 94

A multi-party PPML framework - Paillier's PHE encryption & Federated learning

- Study of the research efforts published in [1].
- The authors propose, a multi-party PPML framework, that is based on
  - partially homomorphic encryption using an improved Paillier algorithm to speed up the training by 25–28%
  - federated learning
  - a secret multi-party computation protocol

[1] Fang, H.; Qian, Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet 2021, 13, 94

A multi-party PPML framework - Paillier's PHE encryption & Federated learning

- Study of the research efforts published in [1].
- The authors propose, a multi-party PPML framework, that is based on
  - partially homomorphic encryption using an improved Paillier algorithm to speed up the training by 25–28%
  - federated learning
  - a secret multi-party computation protocol
- The core idea is based on ensuring that all learning parties just transmitting the encrypted gradients by homomorphic encryption.

[1] Fang, H.; Qian, Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet 2021, 13, 94

Paillier Federated Network

Paillier Federated Network

Motivation

- As such a federated network is used to allow multiple parties to perform cooperative ML with *isolated data*.

[1] Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, 12–16 October 2015; pp. 1310–1321.

Paillier Federated Network

Motivation

- As such a federated network is used to allow multiple parties to perform cooperative ML with *isolated data*.
- However, in practice, is there a guarantee that not only the data provided by the participants is safe, but also the final model trained by multiple parties?

[1] Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, 12–16 October 2015; pp. 1310–1321.

Paillier Federated Network

Motivation

- As such a federated network is used to allow multiple parties to perform cooperative ML with *isolated data*.
- However, in practice, is there a guarantee that not only the data provided by the participants is safe, but also the final model trained by multiple parties?
- There cannot be such assurances.

[1] Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, 12–16 October 2015; pp. 1310–1321.

Paillier Federated Network

Motivation

- As such a federated network is used to allow multiple parties to perform cooperative ML with *isolated data*.
- However, in practice, is there a guarantee that not only the data provided by the participants is safe, but also the final model trained by multiple parties?
- There cannot be such assurances.
- In fact, there have been attacks proposed in the state-of-the-art

[1] Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, 12–16 October 2015; pp. 1310–1321.

Paillier Federated Network

Motivation

- As such a federated network is used to allow multiple parties to perform cooperative ML with *isolated data*.
- However, in practice, is there a guarantee that not only the data provided by the participants is safe, but also the final model trained by multiple parties?
- There cannot be such assurances.
- In fact, there have been attacks proposed in the state-of-the-art
  - a member inference attack[1] discusses that a cracker can invade the server and infer several shadow models from the data in the server.

[1] Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, 12–16 October 2015; pp. 1310–1321.

Paillier Federated Network

Motivation

- As such a federated network is used to allow multiple parties to perform cooperative ML with *isolated data*.
- However, in practice, is there a guarantee that not only the data provided by the participants is safe, but also the final model trained by multiple parties?
- There cannot be such assurances.
- In fact, there have been attacks proposed in the state-of-the-art
  - a member inference attack[1] discusses that a cracker can invade the server and infer several shadow models from the data in the server.
    - Eventually, can get a prediction that is similar to the model trained by actual cooperation.

[1] Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, 12–16 October 2015; pp. 1310–1321.

Paillier Federated Network

Motivation

- As such a federated network is used to allow multiple parties to perform cooperative ML with *isolated data*.
- However, in practice, is there a guarantee that not only the data provided by the participants is safe, but also the final model trained by multiple parties?
- There cannot be such assurances.
- In fact, there have been attacks proposed in the state-of-the-art
  - a member inference attack[1] discusses that a cracker can invade the server and infer several shadow models from the data in the server.
    - Eventually, can get a prediction that is similar to the model trained by actual cooperation.
  - in addition, is it possible to always ensure a trusted server?

[1] Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, 12–16 October 2015; pp. 1310–1321.

Paillier Federated Network

Motivation

- As such a federated network is used to allow multiple parties to perform cooperative ML with *isolated data*.
- However, in practice, is there a guarantee that not only the data provided by the participants is safe, but also the final model trained by multiple parties?
- There cannot be such assurances.
- In fact, there have been attacks proposed in the state-of-the-art
  - a member inference attack[1] discusses that a cracker can invade the server and infer several shadow models from the data in the server.
    - Eventually, can get a prediction that is similar to the model trained by actual cooperation.
  - in addition, is it possible to always ensure a trusted server?
- Hence, the recourse is to use homomorphic encryption...

[1] Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, 12–16 October 2015; pp. 1310–1321.

Paillier Federated Network

Motivation

- As such a federated network is used to allow multiple parties to perform cooperative ML with *isolated data*.
- However, in practice, is there a guarantee that not only the data provided by the participants is safe, but also the final model trained by multiple parties?
- There cannot be such assurances.
- In fact, there have been attacks proposed in the state-of-the-art
  - a member inference attack[1] discusses that a cracker can invade the server and infer several shadow models from the data in the server.
    - Eventually, can get a prediction that is similar to the model trained by actual cooperation.
  - in addition, is it possible to always ensure a trusted server?
- Hence, the recourse is to use homomorphic encryption...

[1] Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, 12–16 October 2015; pp. 1310–1321.

Paillier Federated Network

Motivation

- As such a federated network is used to allow multiple parties to perform cooperative ML with *isolated data*.
- However, in practice, is there a guarantee that not only the data provided by the participants is safe, but also the final model trained by multiple parties?
- There cannot be such assurances.
- In fact, there have been attacks proposed in the state-of-the-art
  - a member inference attack[1] discusses that a cracker can invade the server and infer several shadow models from the data in the server.
    - Eventually, can get a prediction that is similar to the model trained by actual cooperation.
  - in addition, is it possible to always ensure a trusted server?
- Hence, the recourse is to use homomorphic encryption...

The paper by Fang H et al proposes to use Paillier's additively homomorphic encrytion algorithm for the purpose.

[1] Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, Colorado, 12–16 October 2015; pp. 1310–1321.

Paillier Partially Additively Homomorphic Encryption Algorithm

Algorithm Paillier ()

Key Generation:

1. Choose two large prime numbers p and q randomly and independently of each other such that gcd(pq,(p-1)(q-1))=1.
2. This property is assured if both primes are of equivalent length, i.e. $p, q \in 1||\{0,1\}^{\{s-1\}}$ for security parameter s.
3. Compute n=pq and $\lambda = lcm(p-1, q-1)$.
4. Select random integer g where $g \in Z^*_{n^2}$.
   Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse:
   $\mu = (L(g^{\lambda} mod\, n^2))^{-1} mod\, n$,
   where function L is defined as, L(u) = (u-1)/n
5. The public (encryption) key is (n,g).
6. The private (decryption) key is (λ, μ).

Message Encryption: Let m be a message to be encrypted where $m \in Z_n$.
Select a random r where r $\in Z_{n^*}$
Compute ciphertext as: $c = g^m.r^n \ mod\, n^2$

Decryption: Ciphertext c $\in Z^*_{n^2}$
Compute message: $m = L(c^{\lambda} \ mod\, n^2).\mu \ mod\, n$

Figure: Paillier's original AHE algorithm

Improved Paillier Partially Additively Homomorphic Encryption Algorithm

The authors propose a modified Pailliers AHE algorithm. This is in order to improve the efficiency of network training by reducing the complexity.

- Key generation: The authors propose to use α as the divisor. The random number g in the public key is to be of the order αn.

Improved Paillier Partially Additively Homomorphic Encryption Algorithm

The authors propose a modified Pailliers AHE algorithm. This is in order to improve the efficiency of network training by reducing the complexity.

- Key generation: The authors propose to use α as the divisor. The random number g in the public key is to be of the order αn.
- Encryption: Assuming that the plaintext is m, the ciphertext is c, and r is a random positive integer such that $r \leq \alpha$. Then, the improved encryption process be as follows:

$$c = g^m (g^n)^r \ mod \ n^2$$

The authors propose a modified Pailliers AHE algorithm. This is in order to improve the efficiency of network training by reducing the complexity.

- Key generation: The authors propose to use α as the divisor. The random number g in the public key is to be of the order αn.
- Encryption: Assuming that the plaintext is m, the ciphertext is c, and r is a random positive integer such that $r \leq \alpha$. Then, the improved encryption process be as follows:

$$c = g^m (g^n)^r \ mod \ n^2$$

- Decryption: The Decryption process can be shown as:

$$m = \frac{L(c^\alpha \ mod \ n^2)}{L(g^\alpha \ mod \ n^2)} \ mod \ n$$

The authors propose a modified Pailliers AHE algorithm. This is in order to improve the efficiency of network training by reducing the complexity.

- Key generation: The authors propose to use α as the divisor. The random number g in the public key is to be of the order αn.
- Encryption: Assuming that the plaintext is m, the ciphertext is c, and r is a random positive integer such that $r \leq \alpha$. Then, the improved encryption process be as follows:
$$c = g^m (g^n)^r \ mod \ n^2$$
- Decryption: The Decryption process can be shown as:
$$m = \frac{L(c^\alpha \ mod \ n^2)}{L(g^\alpha \ mod \ n^2)} \ mod \ n$$
- Thus, the advantage of using α instead of λ is in the decryption. That is,

The authors propose a modified Pailliers AHE algorithm. This is in order to improve the efficiency of network training by reducing the complexity.

- Key generation: The authors propose to use α as the divisor. The random number g in the public key is to be of the order αn.
- Encryption: Assuming that the plaintext is m, the ciphertext is c, and r is a random positive integer such that $r \leq \alpha$. Then, the improved encryption process be as follows:
$$c = g^m (g^n)^r \ mod \ n^2$$
- Decryption: The Decryption process can be shown as:
$$m = \frac{L(c^\alpha \ mod \ n^2)}{L(g^\alpha \ mod \ n^2)} \ mod \ n$$
- Thus, the advantage of using α instead of λ is in the decryption. That is,
  - the number of power operations has changed from 2λ times to 2α times.

The authors propose a modified Pailliers AHE algorithm. This is in order to improve the efficiency of network training by reducing the complexity.

- Key generation: The authors propose to use α as the divisor. The random number g in the public key is to be of the order αn.
- Encryption: Assuming that the plaintext is m, the ciphertext is c, and r is a random positive integer such that $r \leq \alpha$. Then, the improved encryption process be as follows:

$$c = g^m (g^n)^r \ mod \ n^2$$

- Decryption: The Decryption process can be shown as:

$$m = \frac{L(c^\alpha \ mod \ n^2)}{L(g^\alpha \ mod \ n^2)} \ mod \ n$$

- Thus, the advantage of using α instead of λ is in the decryption. That is,
  - the number of power operations has changed from 2λ times to 2α times.
  - Since α is a divisor of λ, the time overhead has been significantly reduced.

The authors propose a modified Pailliers AHE algorithm. This is in order to improve the efficiency of network training by reducing the complexity.

- Key generation: The authors propose to use α as the divisor. The random number g in the public key is to be of the order αn.
- Encryption: Assuming that the plaintext is m, the ciphertext is c, and r is a random positive integer such that $r \leq \alpha$. Then, the improved encryption process be as follows:
$$c = g^m (g^n)^r \ mod \ n^2$$
- Decryption: The Decryption process can be shown as:
$$m = \frac{L(c^\alpha \ mod \ n^2)}{L(g^\alpha \ mod \ n^2)} \ mod \ n$$
- Thus, the advantage of using α instead of λ is in the decryption. That is,
  - the number of power operations has changed from 2λ times to 2α times.
  - Since α is a divisor of λ, the time overhead has been significantly reduced.
- The computational complexity of Native Paillier is $O(|n|^3)$, and the computational complexity of proposed improved Paillier is $O(|n|^2\alpha)$.

Paillier Federated Network Architecture

- Paillier encryption is used here to protect the gradient data.

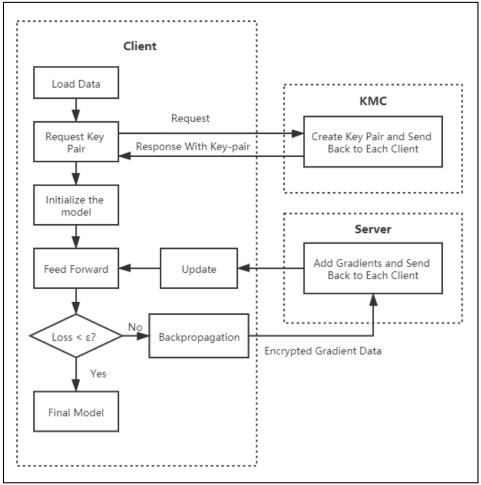

Figure: Paillier Federated Network Architecture

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Paillier Federated Network Architecture

- Paillier encryption is used here to protect the gradient data.
- Therefore, even if the computing server is compromised, the specific information of the gradient data from each learning client is protected.

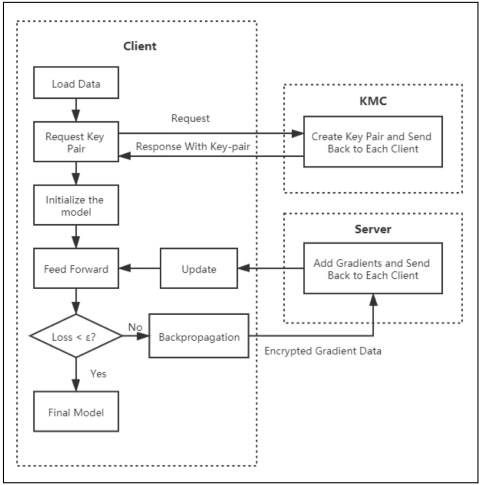

Figure: Paillier Federated Network Architecture

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Paillier Federated Network Architecture

- Paillier encryption is used here to protect the gradient data.
- Therefore, even if the computing server is compromised, the specific information of the gradient data from each learning client is protected.
- In addition, it is impossible for crackers to use these encrypted gradient data to train shadow models.

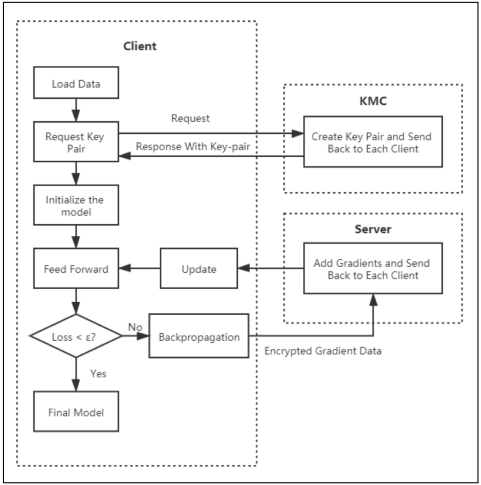

Figure: Paillier Federated Network Architecture

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Paillier Federated Network Architecture

- Paillier encryption is used here to protect the gradient data.
- Therefore, even if the computing server is compromised, the specific information of the gradient data from each learning client is protected.
- In addition, it is impossible for crackers to use these encrypted gradient data to train shadow models.
- What is the purpose of KMC?

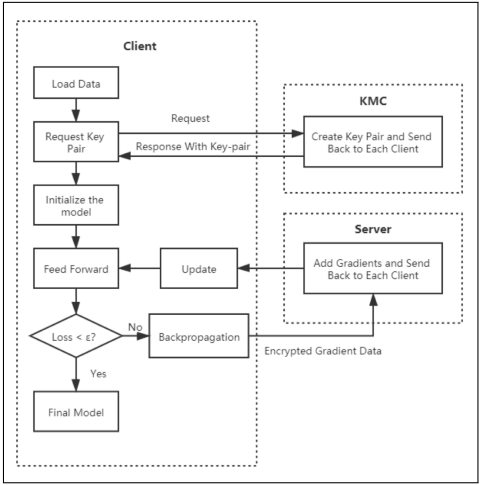

Figure: Paillier Federated Network Architecture

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

Paillier Federated Multi-Layer Perceptron (PFMLP)

Paillier Federated Multi-Layer Perceptron (PFMLP)

- The basic structure of PFMLP is quite similar to FMLP.

Paillier Federated Multi-Layer Perceptron (PFMLP)

- The basic structure of PFMLP is quite similar to FMLP.
- Since PFMLP needs to interact with KMC, the learning client should send a request to the KMC before training starts.

Paillier Federated Multi-Layer Perceptron (PFMLP)

- The basic structure of PFMLP is quite similar to FMLP.
- Since PFMLP needs to interact with KMC, the learning client should send a request to the KMC before training starts.
- The KMC confirms that each participant is online, and then generates key pairs and returns the same to learning clients.

Paillier Federated Multi-Layer Perceptron (PFMLP)

- The basic structure of PFMLP is quite similar to FMLP.
- Since PFMLP needs to interact with KMC, the learning client should send a request to the KMC before training starts.
- The KMC confirms that each participant is online, and then generates key pairs and returns the same to learning clients.
- After getting the key pairs, each learning client performs multi-party machine learning based on encrypted data.

Paillier Federated Multi-Layer Perceptron (PFMLP)

- The basic structure of PFMLP is quite similar to FMLP.
- Since PFMLP needs to interact with KMC, the learning client should send a request to the KMC before training starts.
- The KMC confirms that each participant is online, and then generates key pairs and returns the same to learning clients.
- After getting the key pairs, each learning client performs multi-party machine learning based on encrypted data.
- The flow of operations in PFMLP is shown in the next diagram. PFMLP, contains learning clients, computing server, and KMC.

Paillier Federated Multi-Layer Perceptron (PFMLP)

- The basic structure of PFMLP is quite similar to FMLP.
- Since PFMLP needs to interact with KMC, the learning client should send a request to the KMC before training starts.
- The KMC confirms that each participant is online, and then generates key pairs and returns the same to learning clients.
- After getting the key pairs, each learning client performs multi-party machine learning based on encrypted data.
- The flow of operations in PFMLP is shown in the next diagram. PFMLP, contains learning clients, computing server, and KMC.
- That is, as compared to FMLP, PFMLP adds another three parts:

Paillier Federated Multi-Layer Perceptron (PFMLP)

- The basic structure of PFMLP is quite similar to FMLP.
- Since PFMLP needs to interact with KMC, the learning client should send a request to the KMC before training starts.
- The KMC confirms that each participant is online, and then generates key pairs and returns the same to learning clients.
- After getting the key pairs, each learning client performs multi-party machine learning based on encrypted data.
- The flow of operations in PFMLP is shown in the next diagram. PFMLP, contains learning clients, computing server, and KMC.
- That is, as compared to FMLP, PFMLP adds another three parts:
  - Encryption and decryption operations in the learning clients

Paillier Federated Multi-Layer Perceptron (PFMLP)

- The basic structure of PFMLP is quite similar to FMLP.
- Since PFMLP needs to interact with KMC, the learning client should send a request to the KMC before training starts.
- The KMC confirms that each participant is online, and then generates key pairs and returns the same to learning clients.
- After getting the key pairs, each learning client performs multi-party machine learning based on encrypted data.
- The flow of operations in PFMLP is shown in the next diagram. PFMLP, contains learning clients, computing server, and KMC.
- That is, as compared to FMLP, PFMLP adds another three parts:
  - Encryption and decryption operations in the learning clients
  - Homomorphic operations in the computing server; and

Paillier Federated Multi-Layer Perceptron (PFMLP)

- The basic structure of PFMLP is quite similar to FMLP.
- Since PFMLP needs to interact with KMC, the learning client should send a request to the KMC before training starts.
- The KMC confirms that each participant is online, and then generates key pairs and returns the same to learning clients.
- After getting the key pairs, each learning client performs multi-party machine learning based on encrypted data.
- The flow of operations in PFMLP is shown in the next diagram. PFMLP, contains learning clients, computing server, and KMC.
- That is, as compared to FMLP, PFMLP adds another three parts:
  - Encryption and decryption operations in the learning clients
  - Homomorphic operations in the computing server; and
  - Generation and distribution of key pairs in the key management center (KMC).

Paillier Federated Multi-Layer Perceptron (PFMLP)

- The basic structure of PFMLP is quite similar to FMLP.
- Since PFMLP needs to interact with KMC, the learning client should send a request to the KMC before training starts.
- The KMC confirms that each participant is online, and then generates key pairs and returns the same to learning clients.
- After getting the key pairs, each learning client performs multi-party machine learning based on encrypted data.
- The flow of operations in PFMLP is shown in the next diagram. PFMLP, contains learning clients, computing server, and KMC.
- That is, as compared to FMLP, PFMLP adds another three parts:
  - Encryption and decryption operations in the learning clients
  - Homomorphic operations in the computing server; and
  - Generation and distribution of key pairs in the key management center (KMC).
- Let us learn all the three operations through the algorithm pseudocode further.

Paillier Federated Network Architecture

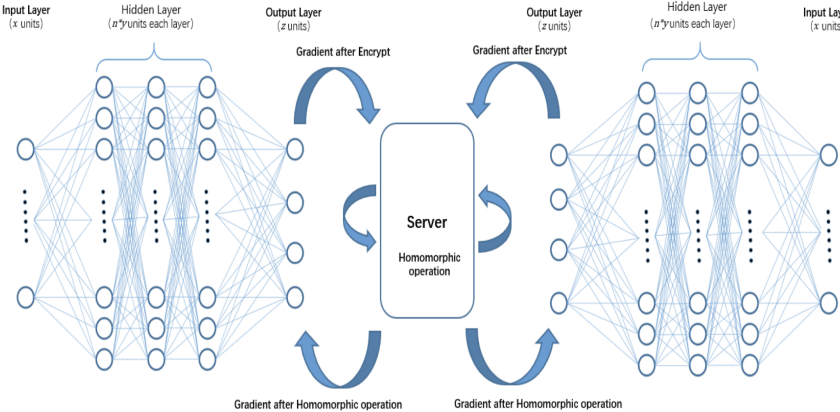

Figure 5. The flow chart of the Paillier federated multi-layer perceptron algorithm.

[1] Figure: Paillier Multilayer Federated Perceptron Architecture

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

PFMLP in the Learning Client

Algorithm 2 PFMLP in the Learning Client

Input: Dataset x
Output: Model θ_{final}

Request key pairs from KMC
Initialize the model parameters θ
for i in *iteration* **do**
    Forward propagation: $out_i = fp(x_i, \theta_i)$;
    Compute loss: $c_i = loss(f^*(x_i), out_i)$;
    if $c_i < \epsilon$ **then**
        Break
    else
        Back propagation: $grad_i = bp(x_i, \theta_i, c_i)$;
        Use public key of client i to encrypt the gradient: $Enc(grad_i) = Enc_{Paillier}(Public_{key}, grad_i)$;
        Send $Enc(grad_i)$ to the computing server and receive: $Enc(grad_{i_{new}})$;
        Use private key of client i to decrypt the gradient: $grad_i = Dec_{Paillier}(Private_{key}, Enc(grad_i))$;
        Update: $\theta_{i+1} = \theta_i - lr * grad_{new}$;
    endif
endfor
return the model with parameters θ_{final};

#	Parameter	Meaning
1	x	the sample in the dataset
2	θ	the parameters of the model
3	f_p	feed forward process
4	*out*	the output of each iteration
5	$f*$	activation function[1]
6	*loss*	loss function
7	c	loss calculated by loss function
8	ϵ	minimum error
9	b_p	back-propagation process
10	*grad*	gradient calculated by bp process
11	l_r	learning rate

1

Figure: Paillier Multilayer Federated Perceptron Architecture

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

PFMLP in the KMC

Algorithm 3 PFMLP in KMC

Input: *requests*
Output: *KeyPair*

```
1: while listening request from Clients do
2:     if receive a request from a client then
3:         Generate a KeyPair;
4:         return a KeyPair to the learning client;
5:     endif
6: endwhile
```

Figure: Paillier Multilayer Federated Perceptron Architecture

1

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

PFMLP in the Computing Server

Algorithm 4 PFMLP in the Computing Server

Input: *requests*
Output: *GradientData*

```
while listening requests from Clients do
    Initialize GradientData;
    if receive a request then
        Push Encrypted Data Enc(data) to the Queue;
        if the requests number == learning clients then
            for i in the number of learning clients do
                GradientData = GradientData ⊕ Enc(data_i)
            endfor
            return GradientData to each client;
            Break;
        endif
    endif
endwhile
```

Figure: Paillier Multilayer Federated Perceptron Architecture[1]

[1] Source: Fang H, Qian Q. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. Future Internet. 2021; 13(4):94.

PFMLP: Security Analysis

- X

ML specific approaches for Privacy Preservation: Ensemble Learning

Ensemble Learning

- is a general meta approach to machine learning that seeks better predictive performance by combining the predictions from multiple models.

....continued

Ensemble Learning

- is a general meta approach to machine learning that seeks better predictive performance by combining the predictions from multiple models.
- the motivation is that it's important to consider and trial more than one algorithm to find the best model for data - akin to listening to more than one opinion on an important matter.

....continued

Ensemble Learning

- is a general meta approach to machine learning that seeks better predictive performance by combining the predictions from multiple models.
- the motivation is that it's important to consider and trial more than one algorithm to find the best model for data - akin to listening to more than one opinion on an important matter.
- by combining the output of different models (instead of relying on a single estimate), ensemble modeling helps to build a consensus on the meaning of the data.

....continued

Ensemble Learning

- is a general meta approach to machine learning that seeks better predictive performance by combining the predictions from multiple models.
- the motivation is that it's important to consider and trial more than one algorithm to find the best model for data - akin to listening to more than one opinion on an important matter.
- by combining the output of different models (instead of relying on a single estimate), ensemble modeling helps to build a consensus on the meaning of the data.
- the aggregated estimates are generally more accurate than any one technique.

....continued

Ensemble Learning

- there are three methods employed to develop a number of ensembles for predictive modeling problem viz.

Ensemble Learning

- there are three methods employed to develop a number of ensembles for predictive modeling problem viz.
  - bagging - involves fitting many decision trees on different samples of the same dataset and averaging the predictions.

Ensemble Learning

- there are three methods employed to develop a number of ensembles for predictive modeling problem viz.
  - bagging - involves fitting many decision trees on different samples of the same dataset and averaging the predictions.
  - stacking - involves fitting many different models types on the same data and using another model to learn how to best combine the predictions. and

Ensemble Learning

- there are three methods employed to develop a number of ensembles for predictive modeling problem viz.
  - bagging - involves fitting many decision trees on different samples of the same dataset and averaging the predictions.
  - stacking - involves fitting many different models types on the same data and using another model to learn how to best combine the predictions. and
  - boosting - involves adding ensemble members sequentially that correct the predictions made by prior models and outputs a weighted average of the predictions.

Ensemble Learning

- there are three methods employed to develop a number of ensembles for predictive modeling problem viz.
  - bagging - involves fitting many decision trees on different samples of the same dataset and averaging the predictions.
  - stacking - involves fitting many different models types on the same data and using another model to learn how to best combine the predictions. and
  - boosting - involves adding ensemble members sequentially that correct the predictions made by prior models and outputs a weighted average of the predictions.
- we shall look at a detailed understanding of each method in a later chapter...

Ensemble Learning

- there are three methods employed to develop a number of ensembles for predictive modeling problem viz.
  - bagging - involves fitting many decision trees on different samples of the same dataset and averaging the predictions.
  - stacking - involves fitting many different models types on the same data and using another model to learn how to best combine the predictions. and
  - boosting - involves adding ensemble members sequentially that correct the predictions made by prior models and outputs a weighted average of the predictions.
- we shall look at a detailed understanding of each method in a later chapter...
- how does EL achieve privacy ?

Blank

Blank