Split Neural Network (SplitNN)

Authors:
Otkrist Gupta and Ramesh Raskar (first work, 2017)
Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar (follow-up works)
Massachusetts Institute of Technology

Slides prepared by: Chandra Thapa (July 2019)

- United States Patent Application Publication (Gupta et al., Dec 2017)
- Journal of Network and Computer Applications (Gupta el al., May 2018)
- Follow-up works (Vepakomma et al., Dec 2018 onward – NIPS 2018, ICLR AI 2019)

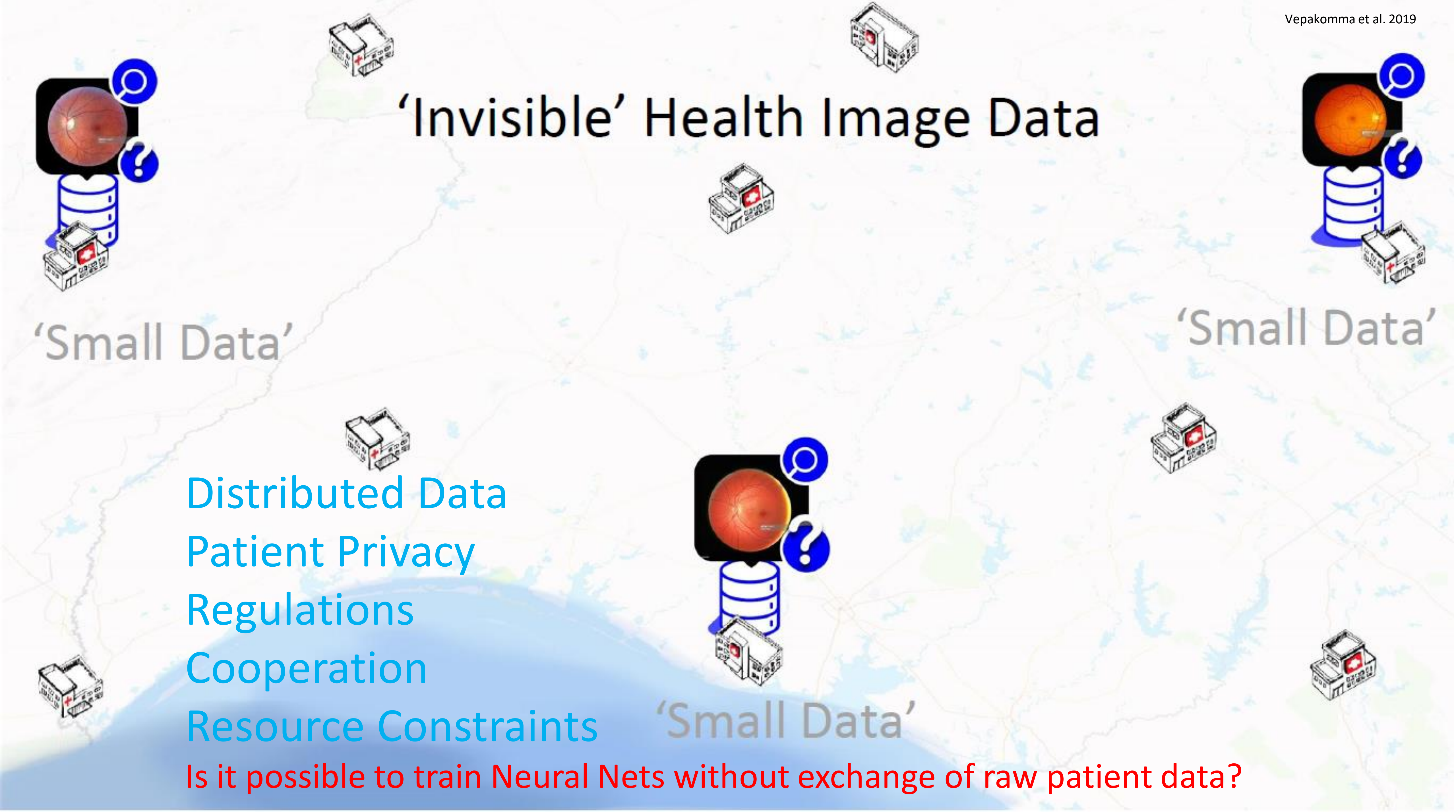
- For resources: https://splitlearning.github.io/

Outline

- Motivations
- SplitNN Algorithms
- Split learning configurations
- Experiments and applications

Vepakomma et al. 2019

‘Invisible’ Health Image Data



Distributed Data

Patient Privacy

Regulations

Cooperation

Resource Constraints

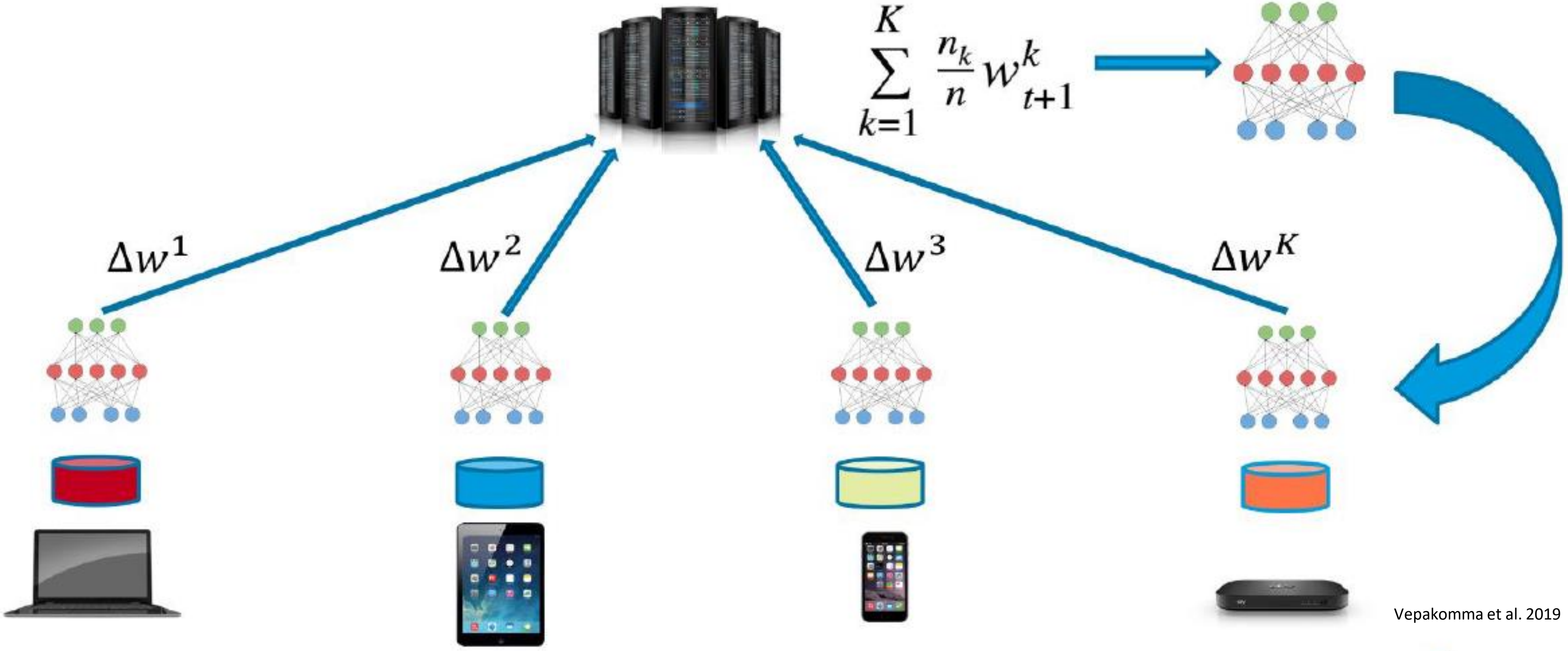
Is it possible to train Neural Nets without exchange of raw patient data?

Federated Learning

McMahan et al. 2017

Server

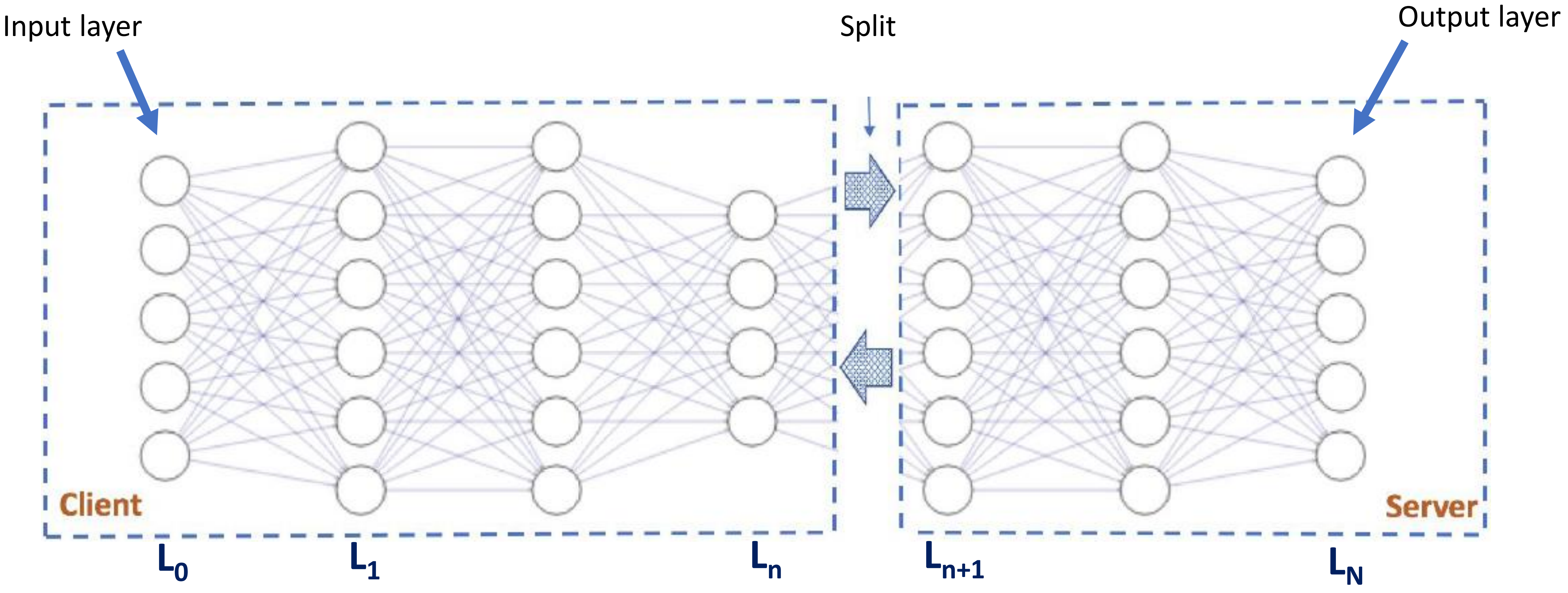
How does it work?



Vepakomma et al. 2019

Vepakomma et al. 2019

SplitNN



Input layer

Split

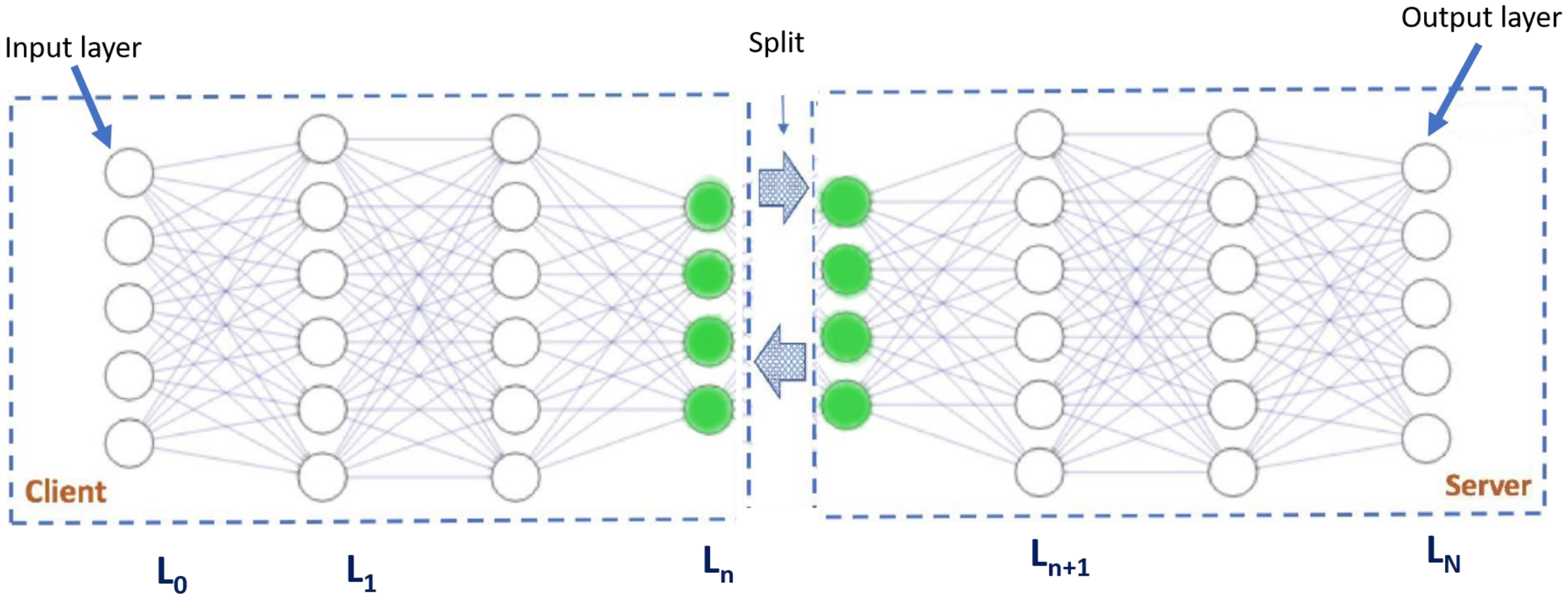
Output layer

L_0 L_1 L_n L_{n+1} L_N

Alice

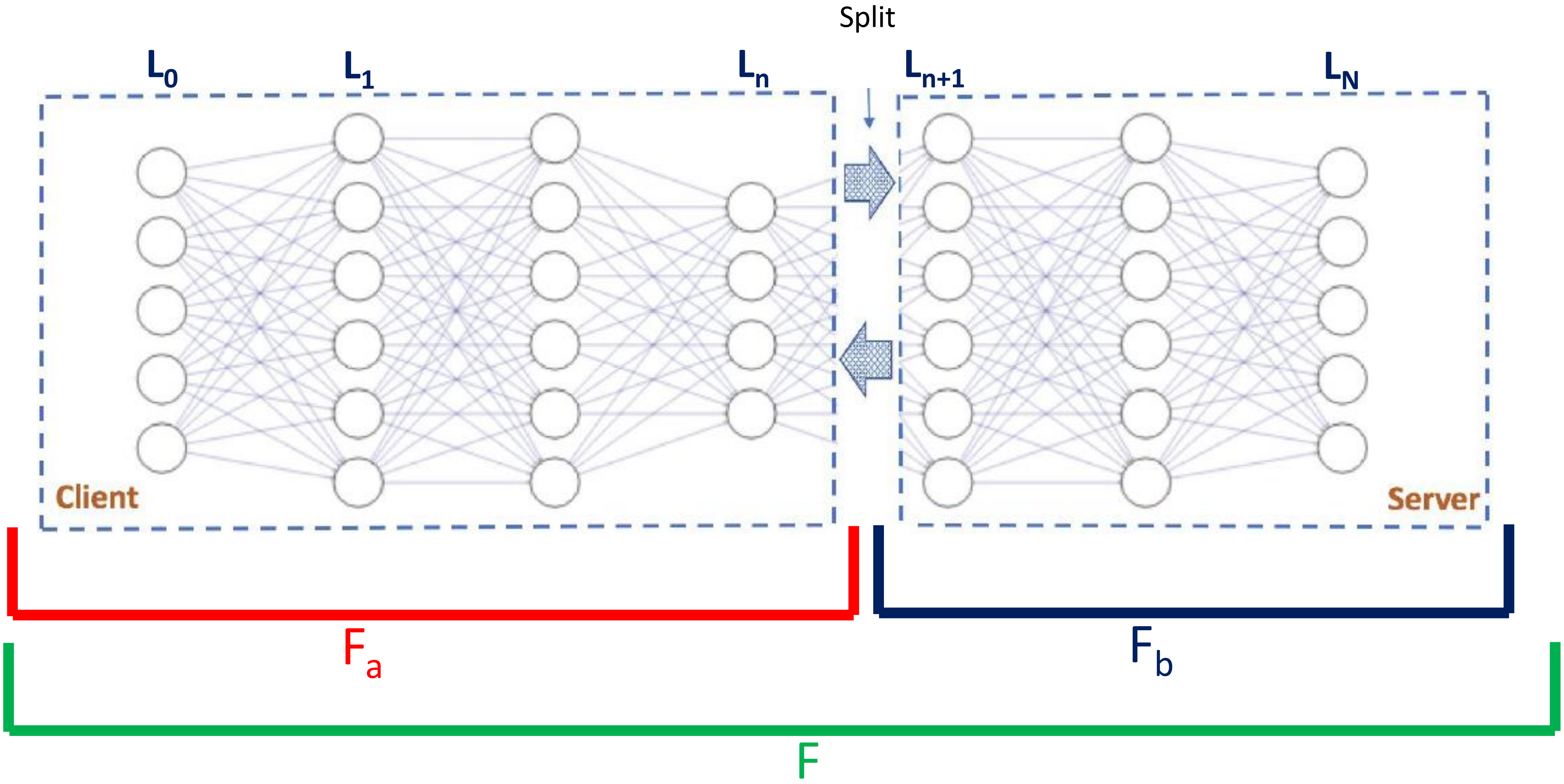
Bob

SplitNN

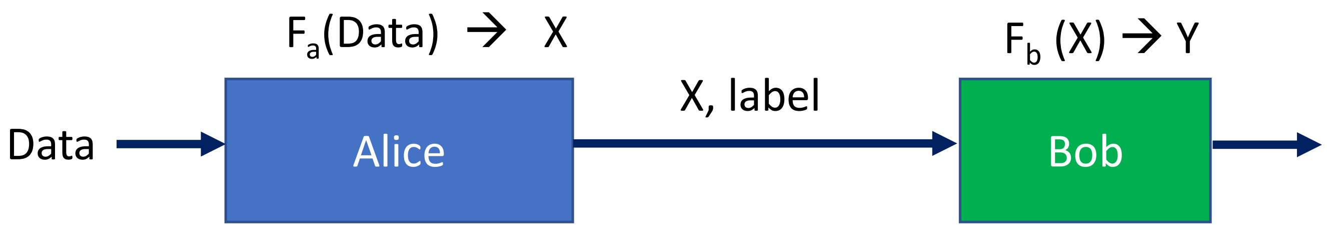


Alice

Bob


Split
L_0
L_1
L_n
L_{n+1}
L_N
Client
Server
F_a
F_b
F

With single Alice



$F_a(Data)$ → X

$F_b(X)$ → Y

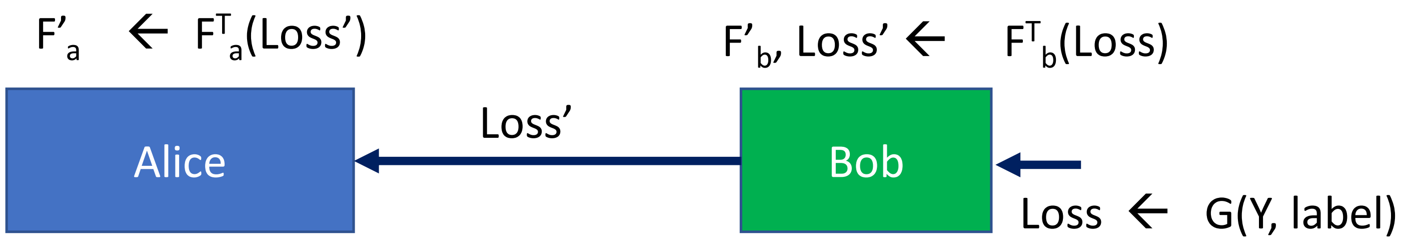
Data

Alice

X, label

Bob

Forward propagation



F'_a ← $F^T_a(Loss')$

F'_b, Loss' ← $F^T_b(Loss)$

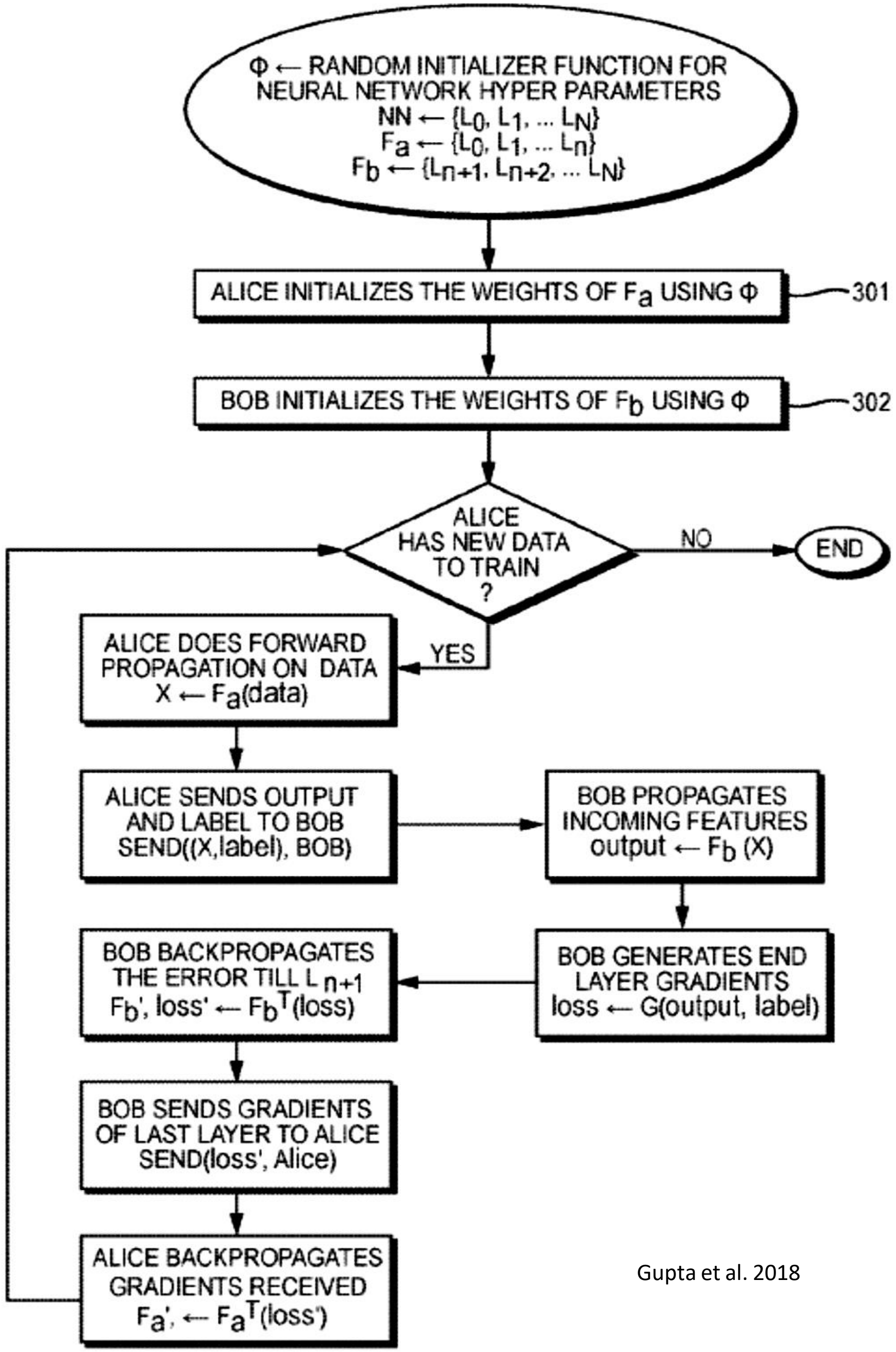
Alice

Loss'

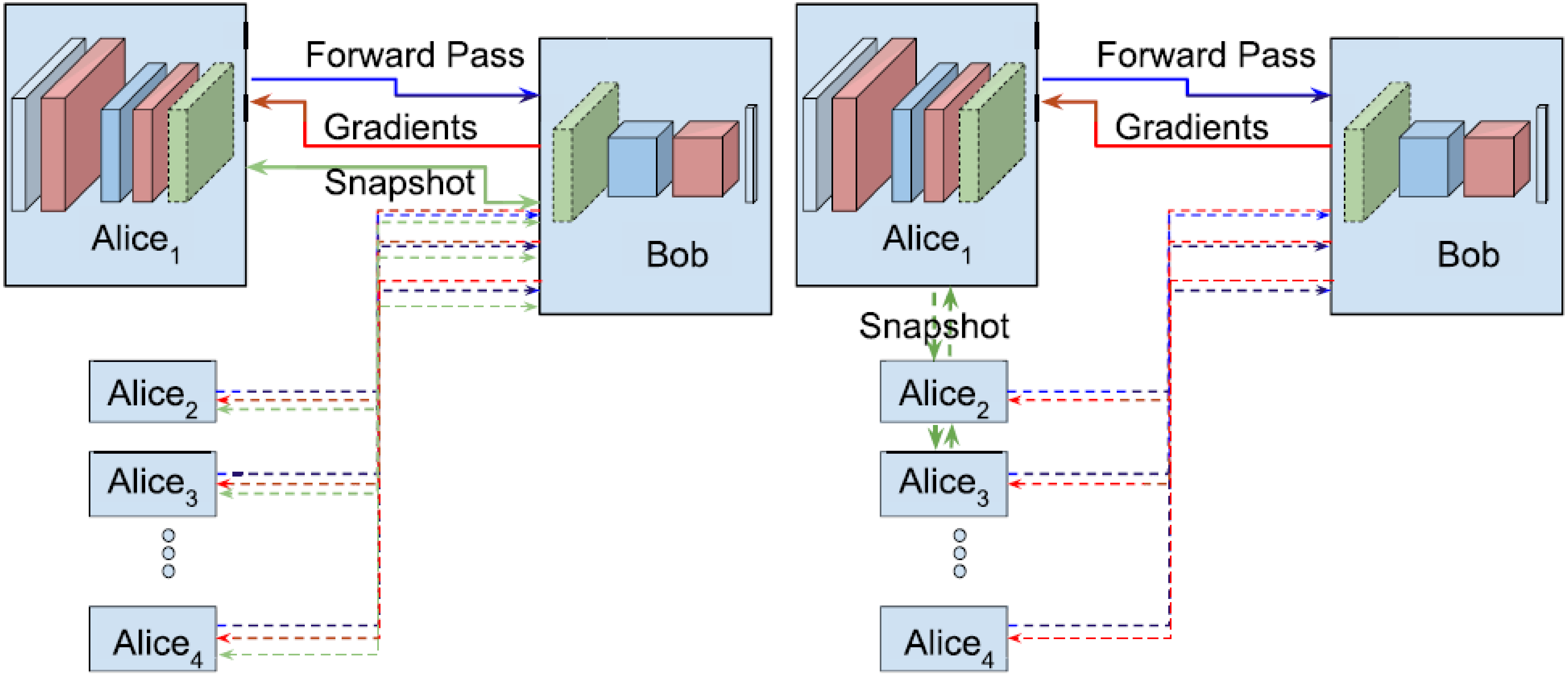
Bob

Loss ← G(Y, label)

Backward propagation


Φ ← RANDOM INITIALIZER FUNCTION FOR NEURAL NETWORK HYPER PARAMETERS
NN ← {L0, L1, ... LN}
Fa ← {L0, L1, ... Ln}
Fb ← {Ln+1, Ln+2, ... LN}
ALICE INITIALIZES THE WEIGHTS OF Fa USING Φ
301
BOB INITIALIZES THE WEIGHTS OF Fb USING Φ
302
ALICE HAS NEW DATA TO TRAIN ?
NO
END
YES
ALICE DOES FORWARD PROPAGATION ON DATA X ← Fa(data)
ALICE SENDS OUTPUT AND LABEL TO BOB SEND((X,label), BOB)
BOB PROPAGATES INCOMING FEATURES output ← Fb (X)
BOB GENERATES END LAYER GRADIENTS loss ← G(output, label)
BOB BACKPROPAGATES THE ERROR TILL L n+1 Fb', loss' ← Fb^T(loss)
BOB SENDS GRADIENTS OF LAST LAYER TO ALICE SEND(loss', Alice)
ALICE BACKPROPAGATES GRADIENTS RECEIVED Fa', ← Fa^T(loss')

Gupta et al. 2018

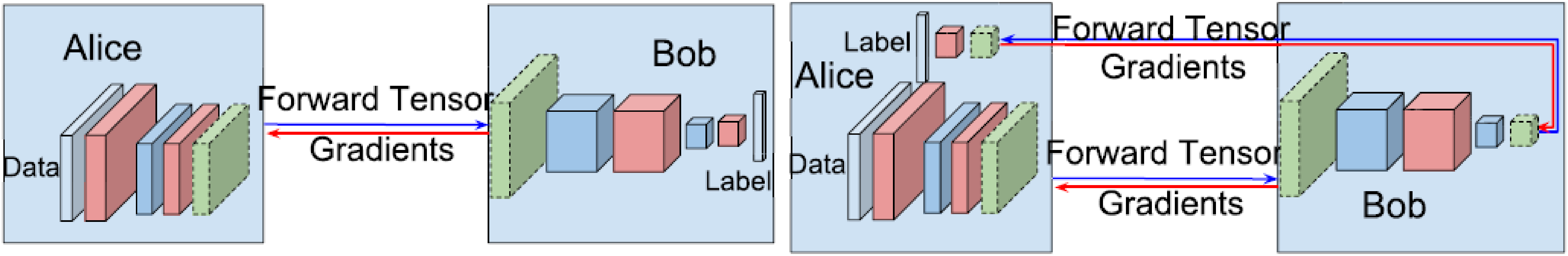


(a) Centralized distributed neural network training.

(b) Peer-to-peer training for distributed learning.

Gupta et al. 2018

Configurations of Split Learning

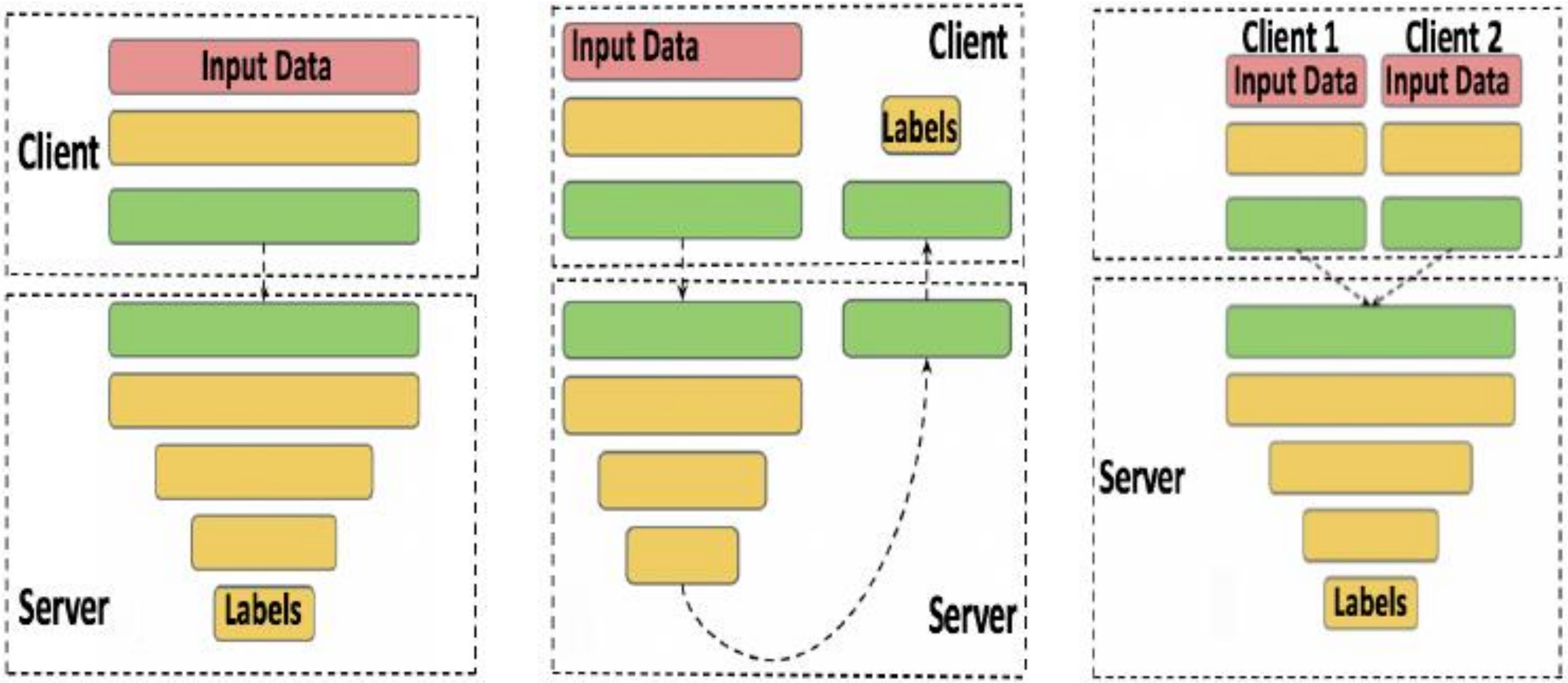


(a) Training with label propagation

(b) Training without label sharing

Gupta et al. 2018

Configurations of Split Learning



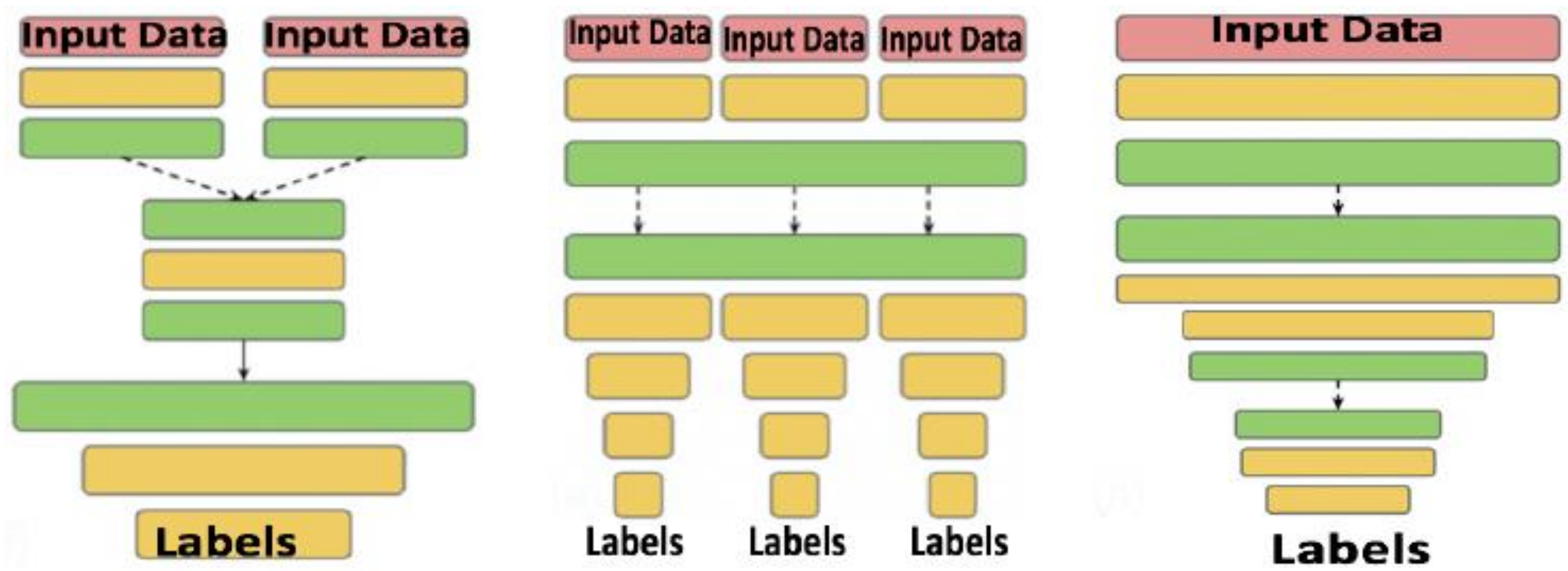
(a) Simple vanilla split learning

(b) Split learning without label sharing

(c) Split learning for vertically partitioned data

Vepakomma et al. 2018

Configurations of Split Learning



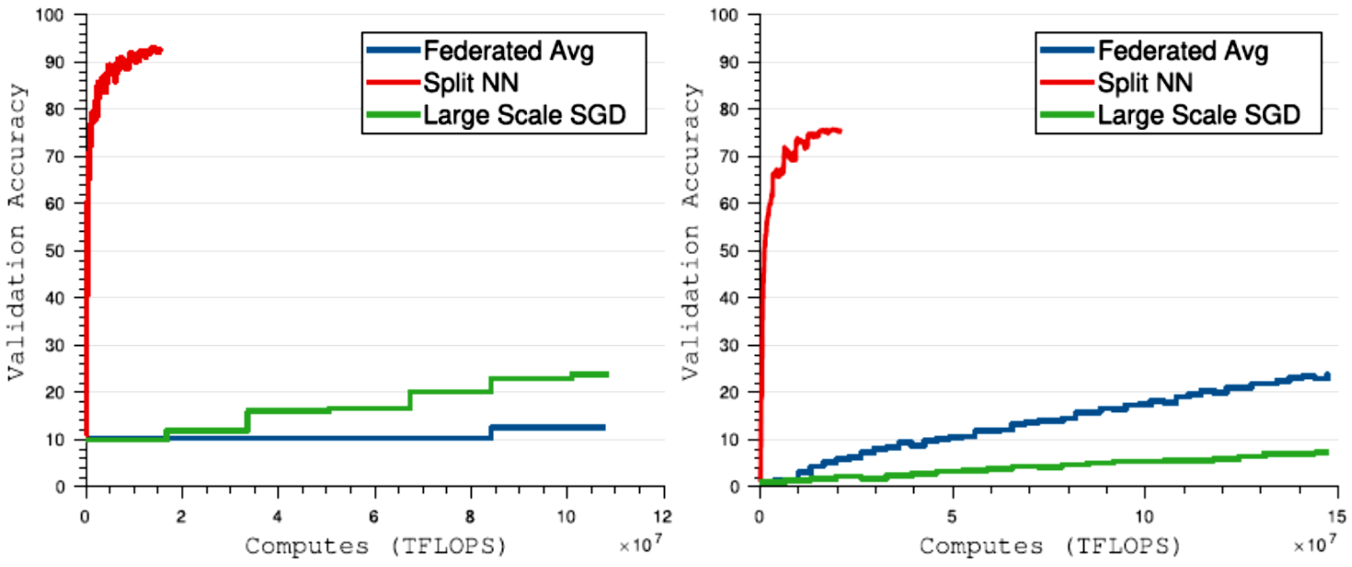
(a) Extended vanilla split learning

(b) Split learning for multi-task output with vertically partitioned input

(c) ’Tor’[28] like multi-hop split learning

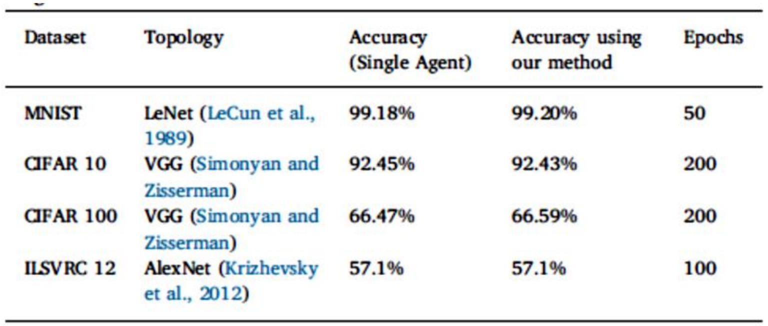
Vepakomma et al. 2018

Results and resource efficiency



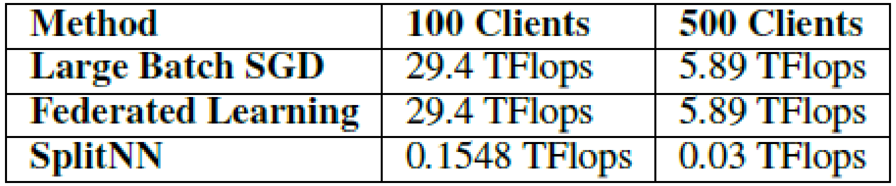
(a) Validation accuracy with client side flops when training 100 clients (VGG and CIFAR 10).

(b) Validation accuracy with client side flops when training 500 clients (Resnet-50 and CIFAR 100).



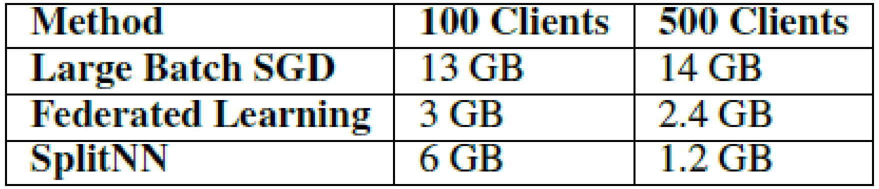
Dataset	Topology	Accuracy (Single Agent)	Accuracy using our method	Epochs
MNIST	LeNet (LeCun et al., 1989)	99.18%	99.20%	50
CIFAR 10	VGG (Simonyan and Zisserman)	92.45%	92.43%	200
CIFAR 100	VGG (Simonyan and Zisserman)	66.47%	66.59%	200
ILSVRC 12	AlexNet (Krizhevsky et al., 2012)	57.1%	57.1%	100

Computational resources consumed per client when training CIFAR 10 over VGG.



Method	100 Clients	500 Clients
Large Batch SGD	29.4 TFlops	5.89 TFlops
Federated Learning	29.4 TFlops	5.89 TFlops
SplitNN	0.1548 TFlops	0.03 TFlops

Communication bandwidth required per client when training CIFAR 100 over ResNet



Method	100 Clients	500 Clients
Large Batch SGD	13 GB	14 GB
Federated Learning	3 GB	2.4 GB
SplitNN	6 GB	1.2 GB

NoPeekNN (Vepakomma et al. 2019)

- Improved SplitNN
  - By minimizing reconstruction of raw data in distributed machine learning by minimizing distance correlation measure between raw data and any intermediary communication between entities while maintaining model accuracies.
  - Reduced leakage during training over colorectal histology image data from 0.92 in traditional CNN and Vanilla SplitNN to 0.33 in NoPeekNN.

Thank You!!