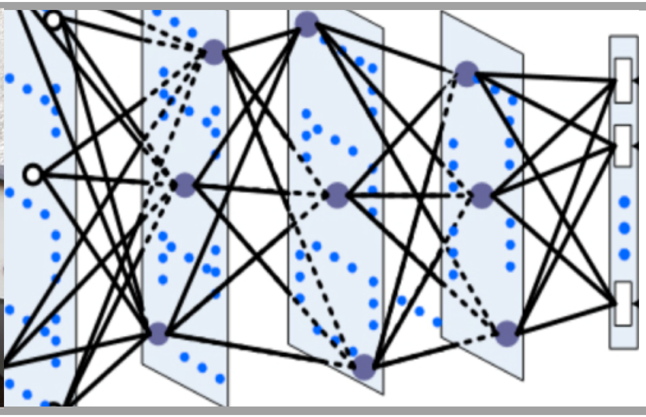
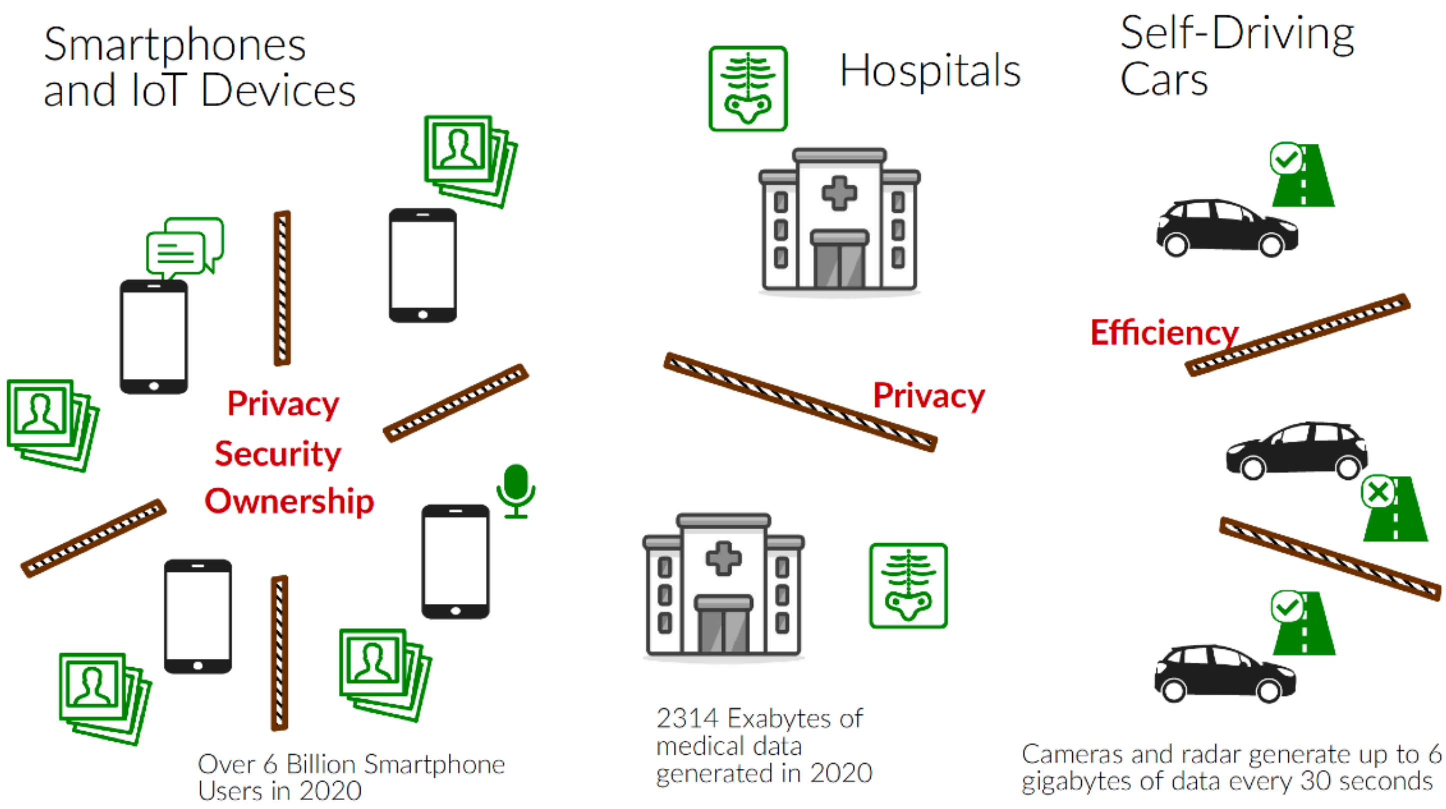
Towards Communication-Efficient and Personalized Federated Learning

Wojciech Samek
Dept. of Artificial Intelligence, Fraunhofer HHI

2021 IEEE SPS Cycle 2 Virtual School on Networked FL: Theory, Algorithms and Applications 31st March 2022

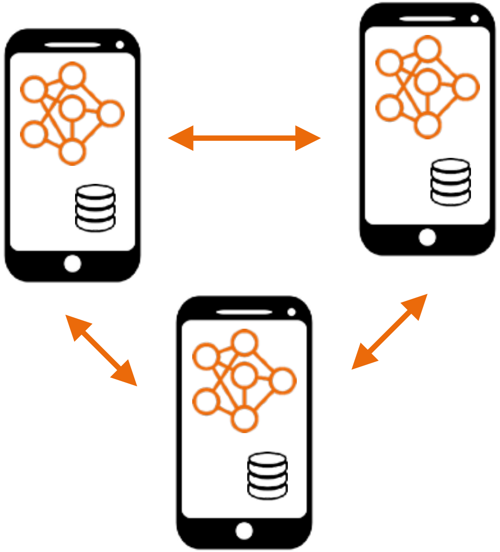
Distributed Data



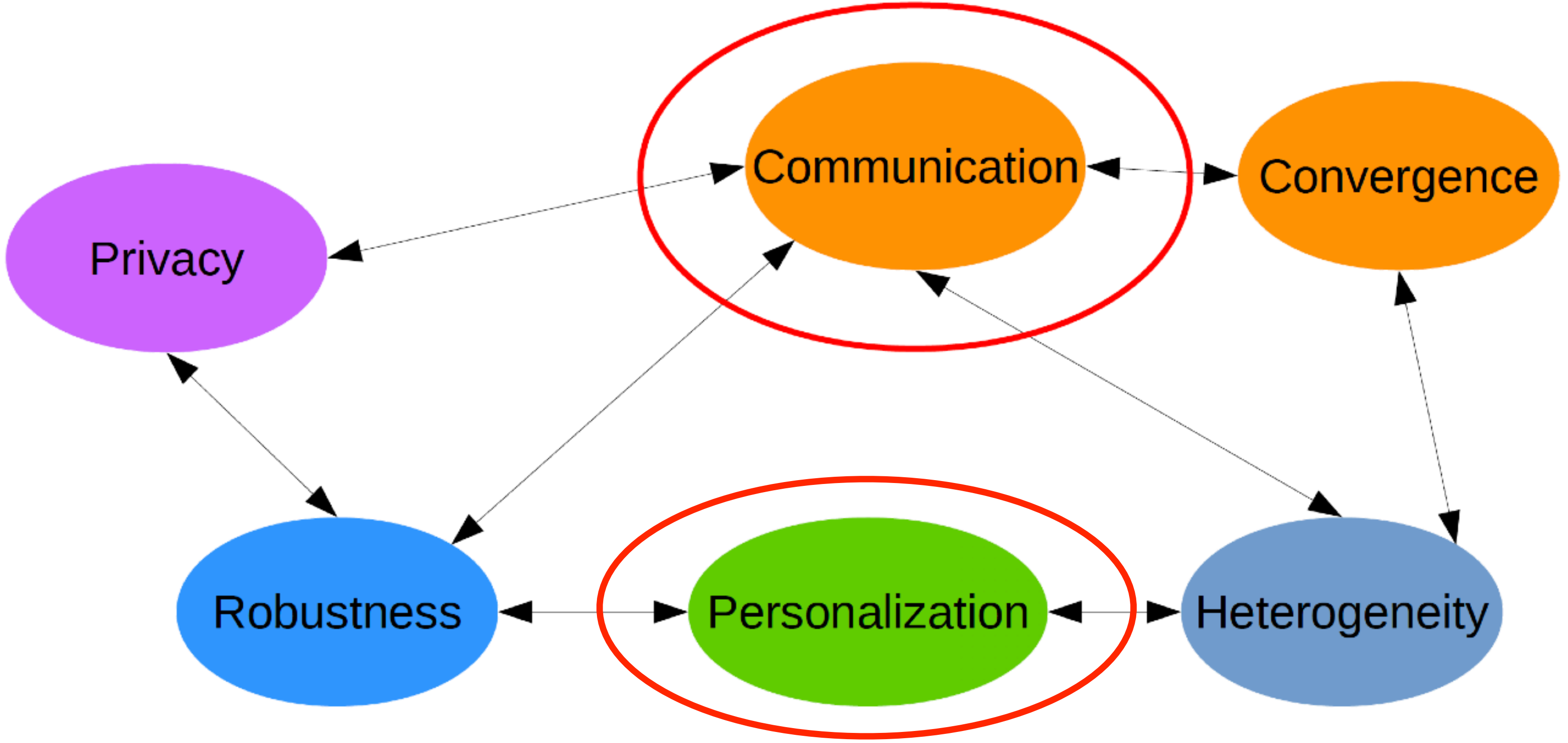
Distributed Data

Centralized Learning

Decentralized Learning

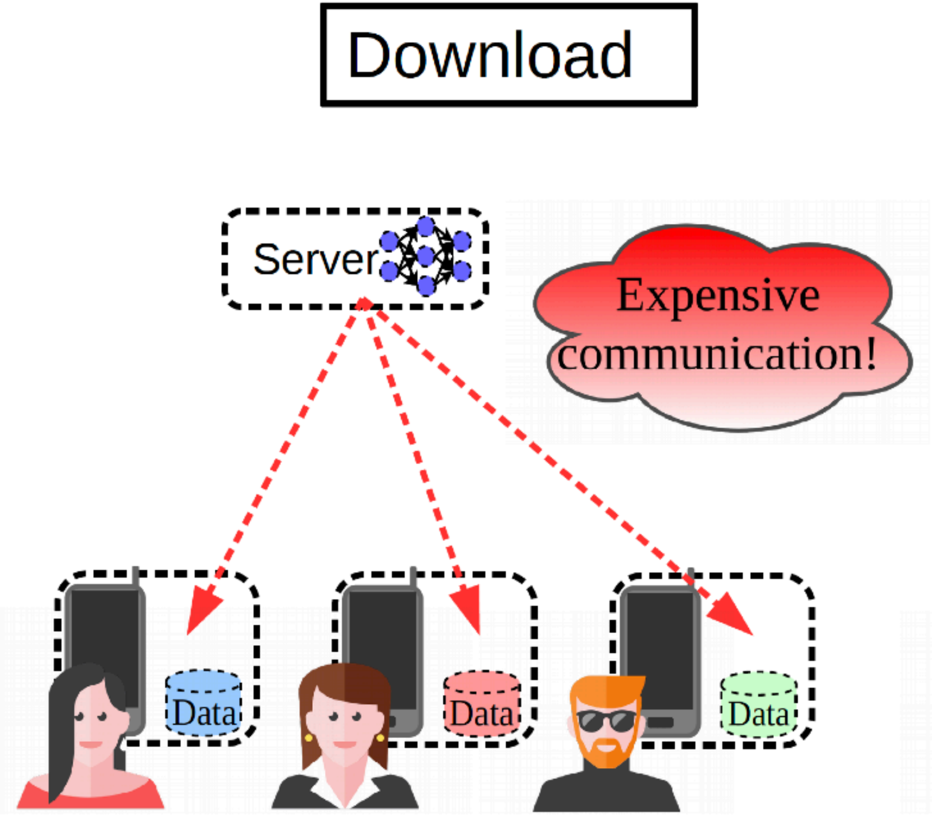


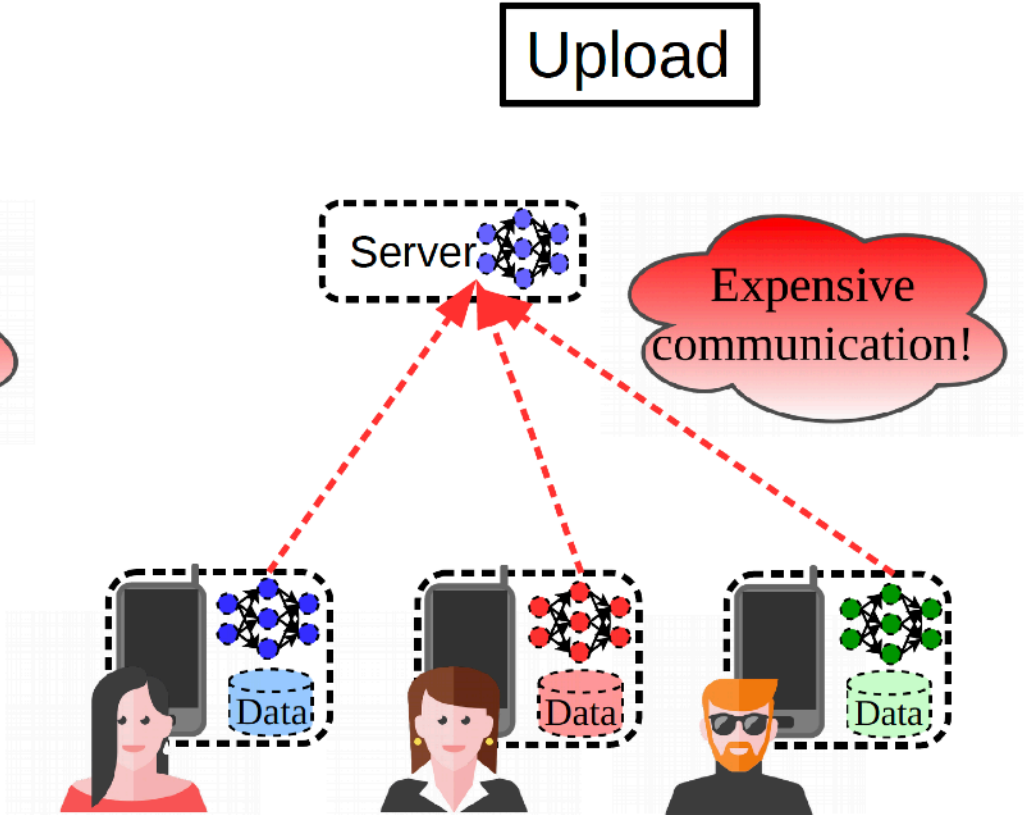
Federated Learning - Challenges



Reducing Communication Overhead

Federated Learning - Challenges





Communication Overhead

Total Communication = [#Communication Rounds] x [#Parameters] x [#Avg. Codeword length]

Communication Overhead (1 client): VGG16 on ImageNet

- Number of Iterations until Convergence: 900.000
- Number of Parameters: 138.000.000
- Bits per Parameters: 32

--> Total Communication = **496.8 Terabyte**

Federated Learning - Compression Methods

Total Communication = [#Communication Rounds] x [#Parameters] x [#Avg. Codeword length]

Compression Methods

- Communication Delay
- Lossy Compression: Unbiased
- Lossy Compression: Biased
- Efficient Encoding

Communication Delay

Distributed SGD:

For t=1,..,[Communication Rounds]:

For i=1,..,[Participating Clients]:

Client does:

$$\boxed{g_i \leftarrow \nabla_\theta l(\theta_t, D_i^b)}$$

Server does:

$$\theta_{t+1} = \theta_t - \eta \frac{1}{M} \sum_i g_i$$

Federated Averaging:

For t=1,..,[Communication Rounds]:

For i=1,..,[Participating Clients]:

Client does:

$$\boxed{\theta_i = \mathrm{SGD}_K(\theta_t, D_i)}$$

Server does:

$$\theta_{t+1} = \frac{1}{M} \sum_i \Delta\theta_i$$

Communication Delay

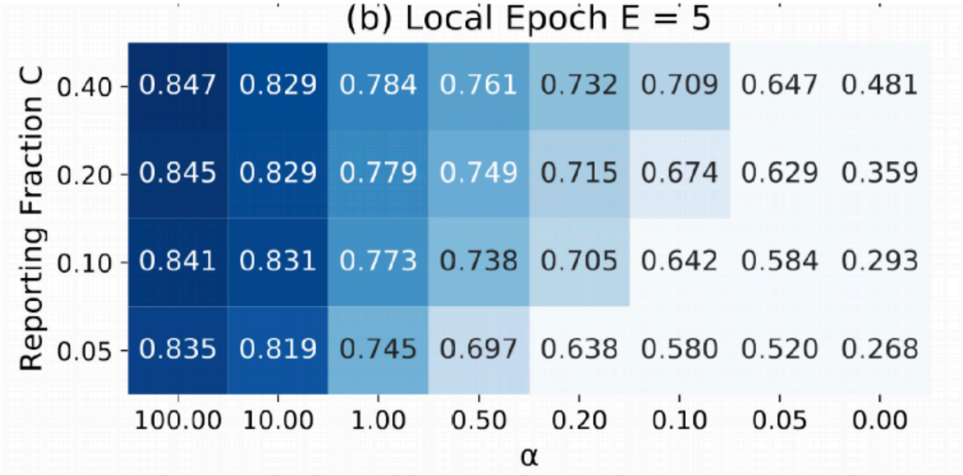
Advantages:

- Simple
- Reduces Communication *Frequency* (advantageous in on-device FL)
- Reduces both Upstream and Downstream communication
- Easy to integrate with Privacy mechanisms

Statistical Heterogeneity

Convergence speed drastically decreases with increasing heterogeneity in the data

→ This effect aggravates if the number of participating clients ("reporting fraction") is low



Dirichlet Parameter

Hsu, Tzu-Ming Harry, Hang Qi, and Matthew Brown. "Measuring the effects of non-identical data distribution for federated visual classification."

Communication Delay

Advantages:

- Simple
- Reduces Communication *Frequency* (more practical in on-device FL)
- Reduces Upstream + Downstream communication
- Easy to integrate with Privacy mechanisms

Disadvantages:

- Bad performance on non-iid data
- Low sample efficiency

Federated Learning - Compression Methods

Total Communication = [#Communication Rounds] x [#Parameters] x [#Avg. Codeword length]

Compression Methods

- Communication Delay
- Lossy Compression: Unbiased
- Lossy Compression: Biased
- Efficient Encoding

Update Compression

Distributed SGD:

For t=1,..,[Communication Rounds]:

For i=1,..,[Participating Clients]:

<u>Client does:</u>

$$\boxed{g_i \leftarrow \nabla_\theta l(\theta_t, D_i^b)}$$

<u>Server does:</u>

$$\theta_{t+1} = \theta_t - \eta \frac{1}{M} \sum_i g_i$$

Distributed SGD with Compression:

For t=1,..,[Communication Rounds]:

For i=1,..,[Participating Clients]:

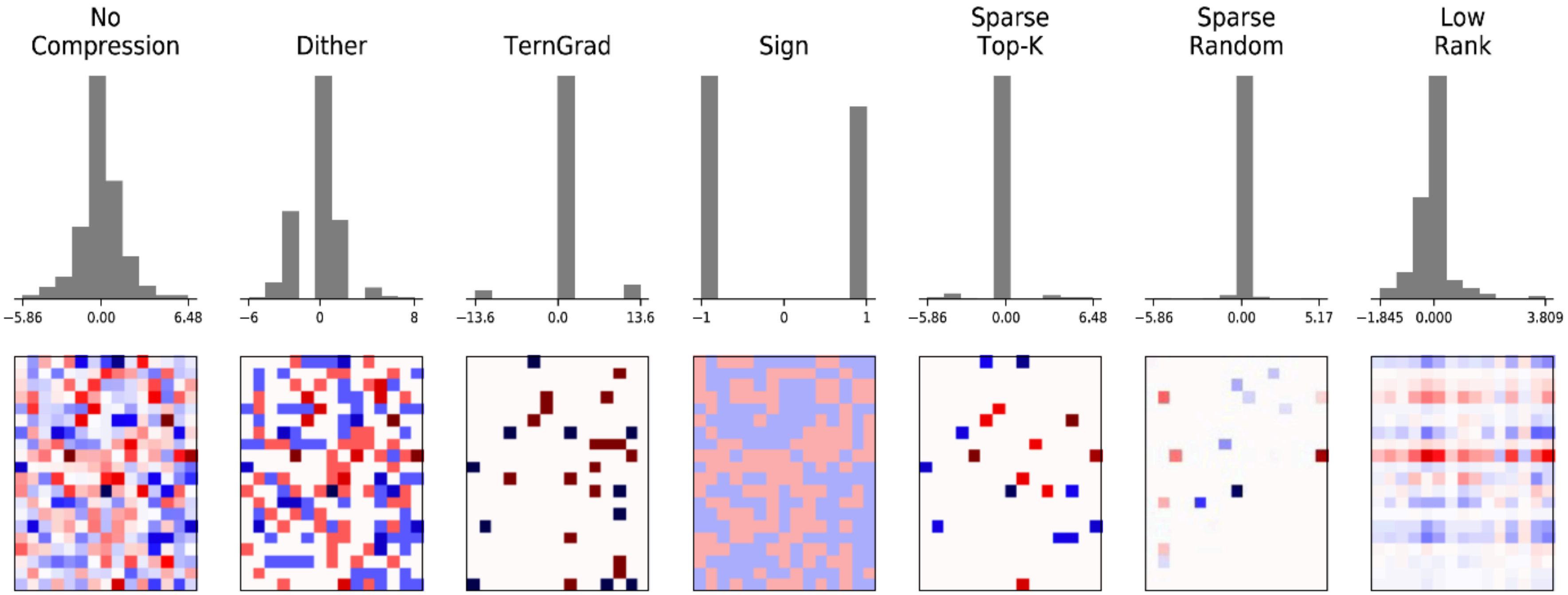
<u>Client does:</u>

$$\boxed{\begin{aligned} g_i &\leftarrow \nabla_\theta l(\theta_t, D_i^b) \\ \tilde{g}_i &\leftarrow \mathrm{comp}(g_i) \end{aligned}}$$

<u>Server does:</u>

$$\theta_{t+1} = \theta_t - \eta \frac{1}{M} \sum_i \tilde{g}_i$$

Update Compression



------ Unbiased ------ --------------------- Biased ---------------------

Update Compression

Algorithm	Test accuracy	Data/epoch	Time per batch	
SGD	94.3%	1023 MB	312 ms	+0%
Atomo	92.6%	113 MB	948 ms	+204%
Signum	93.6%	32 MB	301 ms	−3%
Rank 2	94.4%	8 MB	239 ms	−23%

- Pros: "Straight forward" Convergence Analysis (Stochastic Gradients, increased variance)
- Cons: Variance blow-up leads to poor empirical performance

Biased Compression

Definition: A compression operator $\mathrm{comp}: \mathbb{R}^d \to \mathbb{R}^d$ is called **biased** iff,

$$E[\mathrm{comp}(x)] \neq x \quad \forall x \in \mathbb{R}^d$$

Biased compression methods do not necessarily converge!

→ Can be turned into convergent methods via error accumulation.

Karimireddy, et al. "Error feedback fixes signsgd and other gradient compression schemes."

Stich, Cordonnier, Jaggi. "Sparsified SGD with memory."

Error Accumulation

Distributed SGD:

For t=1,..,[Communication Rounds]:

For i=1,..,[Participating Clients]:

Client does:

$$\boxed{g_i \leftarrow \nabla_\theta l(\theta_t, D_i^b)}$$

Server does:

$$\theta_{t+1} = \theta_t - \eta \frac{1}{M} \sum_i g_i$$

Distributed SGD with Error Accumulation:

For t=1,..,[Communication Rounds]:

For i=1,..,[Participating Clients]:

Client does:

$$\boxed{\begin{aligned} &\mathcal{R}_i \leftarrow \mathcal{R}_i + \nabla_\theta l(\theta_t, D_i^b) \\ &\tilde{g}_i \leftarrow \mathrm{comp}(\mathcal{R}_i) \\ &\mathcal{R}_i \leftarrow \mathcal{R}_i - g_i \end{aligned}}$$

Server does:

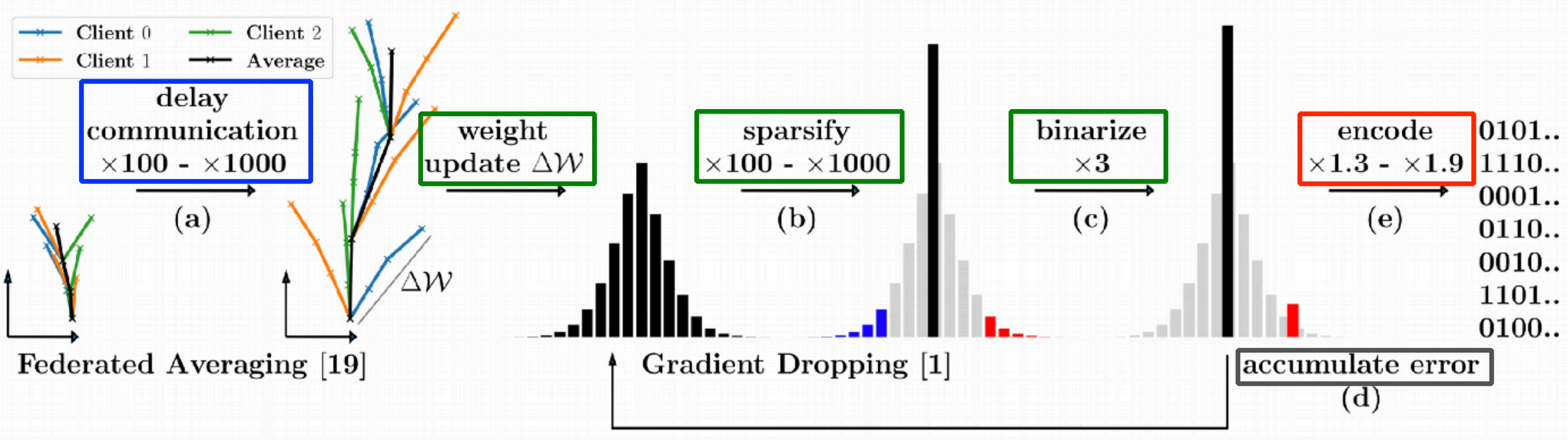
$$\theta_{t+1} = \theta_t - \eta \frac{1}{M} \sum_i \tilde{g}_i$$

Federated Learning - Recap Compression

	Unbiased	Biased
Methods	TernGrad, QSGD, Atomo	Gradient Dropping, Deep Gradient Compression, signSGD, PowerSGD,
Convergence Proofs	Bounded Variance Assumption	k-contraction Framework (Stich et al. 2018)

Combination of Methods: Sparse Binary Compression

Total Communication = [#Communication Rounds] x [#Parameters] x [#Avg. Codeword length]



[Sattler et al. 2019]

Combination of Methods: Sparse Binary Compression

Comm. delay:

$$\Delta\mathcal{W}_i = \mathrm{SGD}_n(\mathcal{W}_i, D_i) - \mathcal{W}_i = [1,\ -2,\ 4,\ 2,\ 1,\ -1]^\top$$

Sparsify & binarize:

$$v^+ = \mathrm{top}_{k\%}(\Delta\mathcal{W}_i) = [4,\ 2] \qquad v^- = \mathrm{top}_{k\%}(-\Delta W) = [2,\ 1]$$

$$\mu^+ = \mathrm{mean}(v^+) = 3 \qquad \mu^- = \mathrm{mean}(v^-) = 1.5$$

$$\Delta\mathcal{W}_i^* = [0,\ 0,\ 3,\ 3,\ 0,\ 0] \longrightarrow$$ send 1 value, 2 positions

Golomb encoding:

Encode distances between nonzero elements (geom. distr.)

$$\mathrm{dist} = [2,\ 0] \qquad \bar{b}_{\mathrm{pos}} = \mathbf{b}^* + 1/(1-p)^{2^{\mathbf{b}^*}}$$

Residual accum.:

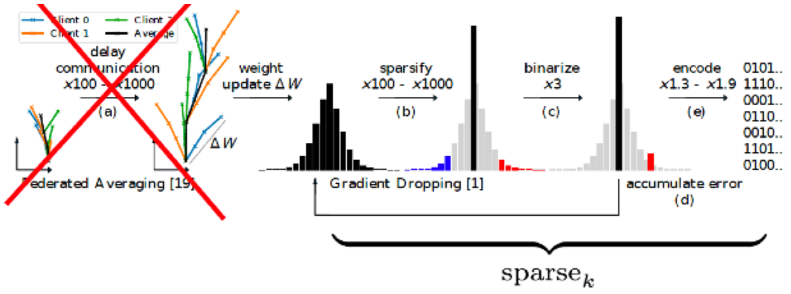
$$R_\tau = R_{\tau-1} + \Delta\mathcal{W}_\tau - \Delta\mathcal{W}_\tau^* = [1,\ -2,\ 1,\ -1,\ 1,\ -1]^\top$$

Results

Compression Method $\longrightarrow$		Baseline	DGC [1]	FedAvg [2]	SBC (1)	SBC (2)	SBC (3)
ResNet50 @ImageNet	Accuracy	0.737	0.739	0.724	0.735	0.737	0.728
	Compression	$\times 1$	$\times 601$	$\times 1000$	$\times 2569$	$\times 3531$	$\times 37208$
ResNet18 @CIFAR10	Accuracy	0.946	0.9383	0.9279	0.9422	0.9435	0.9219
	Compression	$\times 1$	$\times 768$	$\times 1000$	$\times 2369$	$\times 3491$	$\times 31664$

SBC Challenges

- Heterogeneous Data
  - → No communication delay



- Bi-directional Communication
  - → Upstream and Downstream compression

Global Update:

$$\tilde{\Delta\theta}^{(t+1)} = \mathrm{sparse}_k\Big(\frac{1}{n}\sum_{i=1}^{n} \underbrace{\mathrm{sparse}_k(\Delta\theta_i^{(t+1)} + \mathcal{R}_i^{(t)})}_{\tilde{\Delta\theta}_i^{(t+1)}} + \mathcal{R}^{(t)}\Big)$$

- Partial Participation

$$\theta^t \xrightarrow{\tilde{\Delta\theta}^t} \theta^{t+1} \xrightarrow{\tilde{\Delta\theta}^{t+1}} \dots \xrightarrow{\tilde{\Delta\theta}^{T-1}} \theta^T$$

Update Cache:

$$P^{(\tau)} = \{\sum_{t=1}^{s} \tilde{\Delta\theta}^{(T-t)} | s = 1, .., \tau\}$$

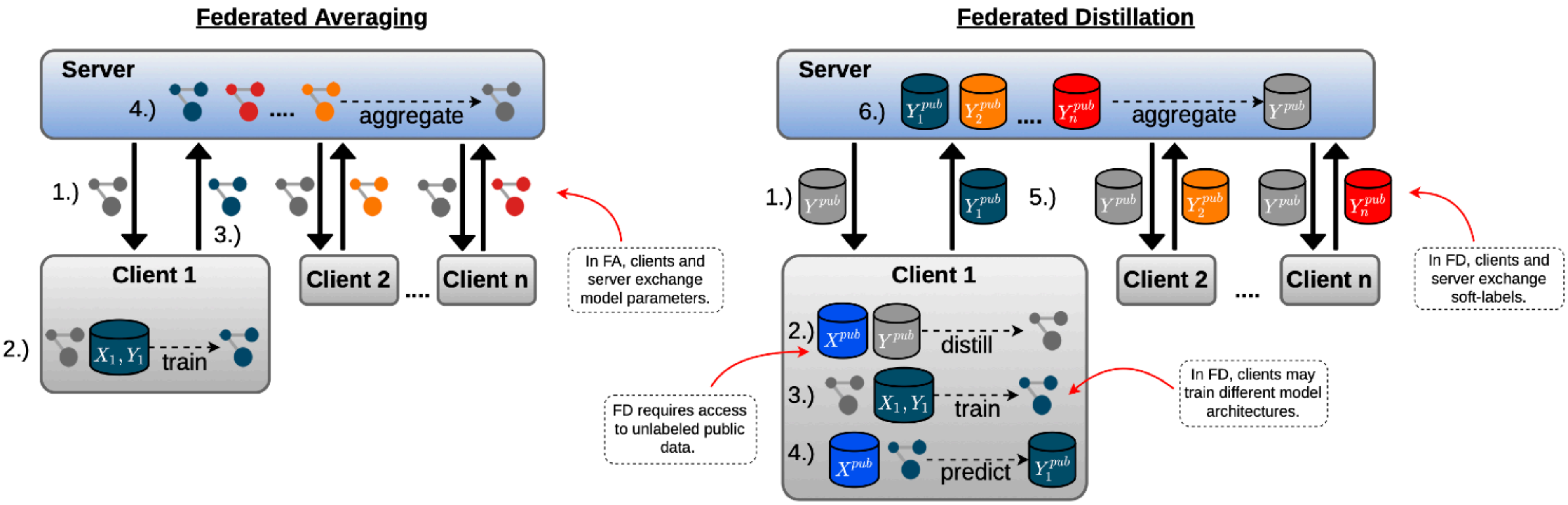
  - → Update Caching

Non-IID Settings

[Sattler et al. 2020]

Communication-Efficient Federated Distillation

Federated Averaging vs. Federated Distillation

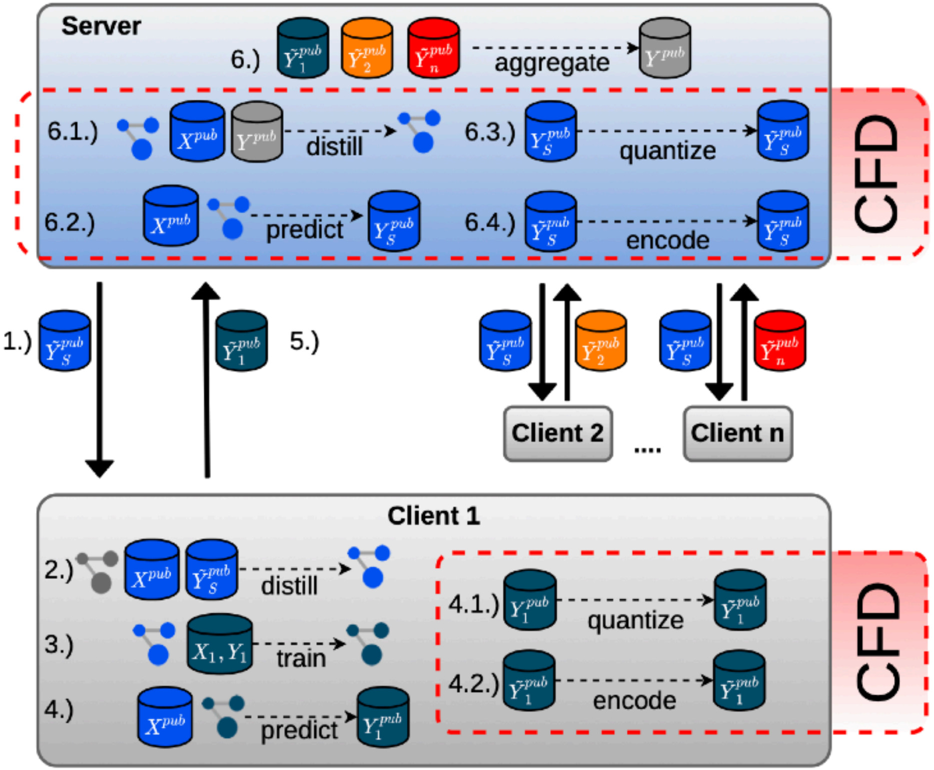


Reducing Communication Overhead in FD

$$\boxed{\mathsf{b}_{total} = |X^{pub}| \times \dim(\mathcal{Y}) \times 32 \text{ bit}}$$

(a) reducing the size of the distillation data set,
(b) reducing the entropy of the soft-labels, or
(c) improving the efficiency of the coding technique.

Compressed Federated Distillation



Distillation Dataset Size

Soft-Label Quantization

$$q = \mathcal{Q}_b(p) = \arg \min_{\substack{q_i \in \{\frac{l}{2^b-1}, l \in 0,..,2^b-1\} \\ \sum_i q_i = 1}} \|q - p\|_1$$

$$\mathcal{Q}_1(p)_i = \begin{cases} 1 & \text{if } i = \arg\max(p) \\ 0 & \text{else} \end{cases}$$

Delta Coding

$$(\hat{Y}^t)_l = \begin{cases} (\tilde{Y}^t)_l & \text{if } (\tilde{Y}^t)_l \neq (\tilde{Y}^{t-1})_l \\ 0 & \text{else} \end{cases}$$

Results

Model	Target Accuracy	α	Up/Down	FA	FD	CFD-1-32	CFD_Δ-1-32	CFD-1-1	CFD_Δ-1-1
ResNet-18	0.71	100.0	up	760.35 (17)	44.80 (14)	0.56 (17)	**0.40** (17)	1.36 (41)	0.82 (41)
			down	760.35 (17)	44.80 (14)	54.40 (17)	54.40 (17)	1.36 (41)	**0.39** (41)
	0.68	1.0	up	1028.71 (23)	48.00 (15)	0.37 (13)	**0.28** (13)	0.64 (22)	0.43 (22)
			down	1028.71 (23)	48.00 (15)	41.60 (13)	41.60 (13)	0.72 (22)	**0.34** (22)
	0.45	0.1	up	1520.70 (34)	16.00 (5)	0.09 (7)	**0.08** (7)	0.52 (41)	0.40 (41)
			down	1520.70 (34)	16.00 (5)	22.40 (7)	22.40 (7)	0.99 (41)	**0.92** (41)
VGG-16	0.8	100.0	up	671.16 (11)	32.00 (10)	0.40 (12)	**0.29** (12)	0.76 (23)	0.47 (23)
			down	671.16 (11)	32.00 (10)	38.40 (12)	38.40 (12)	0.76 (23)	**0.24** (23)
	0.78	1.0	up	1281.30 (21)	28.80 (9)	0.38 (13)	**0.28** (13)	0.56 (19)	0.37 (19)
			down	1281.30 (21)	28.80 (9)	41.60 (13)	41.60 (13)	0.62 (19)	**0.27** (19)
	0.48	0.1	up	2928.69 (48)	25.60 (8)	0.11 (9)	**0.09** (9)	0.43 (34)	0.35 (34)
			down	2928.69 (48)	25.60 (8)	28.80 (9)	28.80 (9)	0.77 (34)	**0.75** (34)
AlexNet	0.68	100.0	up	n.a.	89.60 (28)	0.94 (29)	**0.74** (29)	n.a.	n.a.
			down	n.a.	**89.60** (28)	92.80 (29)	92.80 (29)	n.a.	n.a.
	0.64	1.0	up	n.a.	38.40 (12)	0.61 (21)	**0.49** (21)	0.76 (26)	0.62 (26)
			down	n.a.	38.40 (12)	67.20 (21)	67.20 (21)	0.84 (26)	**0.42** (26)
	0.44	0.1	up	n.a.	6.40 (2)	0.09 (6)	**0.08** (6)	0.11 (7)	0.10 (7)
			down	n.a.	6.40 (2)	19.20 (6)	19.20 (6)	0.17 (7)	**0.15** (7)

[Sattler et al. 2021]

Personalization in FL

Federated Learning - Personalization

Federated Learning Environments are characterized by a **high degree of statistical heterogeneity** of the client data

→ In many situations, learning one single central model is suboptimal or even undesirable

Federated Learning - Personalization

Client data:

$p_i(x, y), \; i = 1, .., n$

$p_i(y|x)$ shared

→ **one model** can be learned

$p_i(y|x)$ varies

→ **no** single model can fit the data of all clients

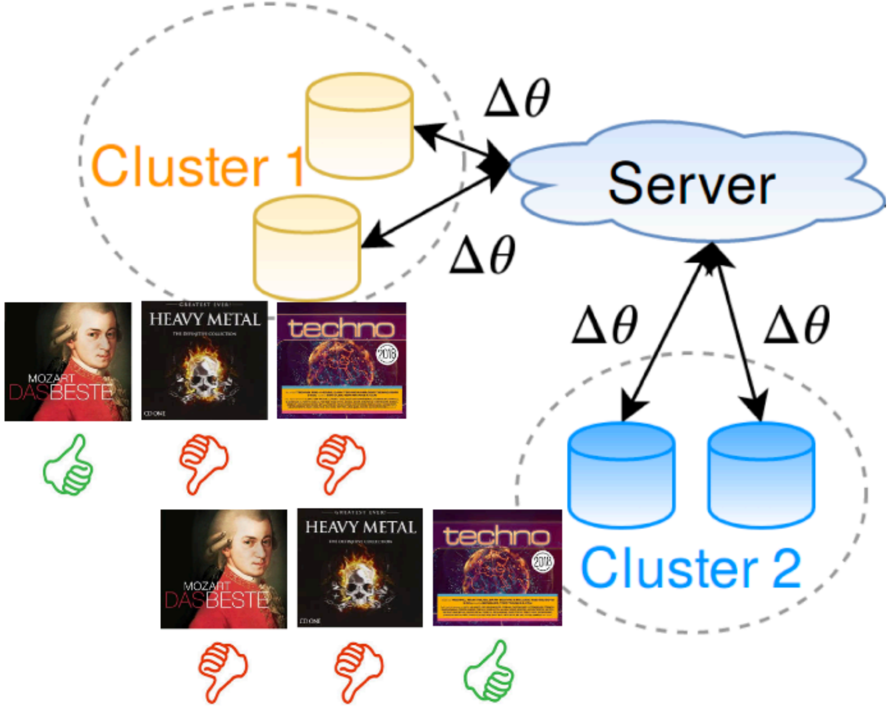
$p_i(x)$ shared

→ IID data

$p_i(x)$ and/or $p_i(y)$ varies

→ non-IID data

Federated Learning - Personalization



A single classifier can not correctly separate the data all clients.

Solution: Use separate model for each cluster

How to identify these clusters ?

Diverging Preferences

Assume: $r_i(\theta) \in \{g(\theta), h(\theta)\}$ → Only k=2 clusters, no intra-cluster variance

local empirical risk

Federated Learning Objective: $F(\theta) := \sum_{i=1}^{n} \frac{|D_i|}{|D|} r_i(\theta) = c_1 g(\theta) + (1-c_1)h(\theta)$

At every stationary solution of the FL objective it holds:

$$0 = \nabla F(\theta^*) = c_1 \nabla g(\theta^*) + (1-c_1)\nabla h(\theta^*)$$

Same preferences

$$0 = \nabla g(\theta^*) = \nabla h(\theta^*)$$

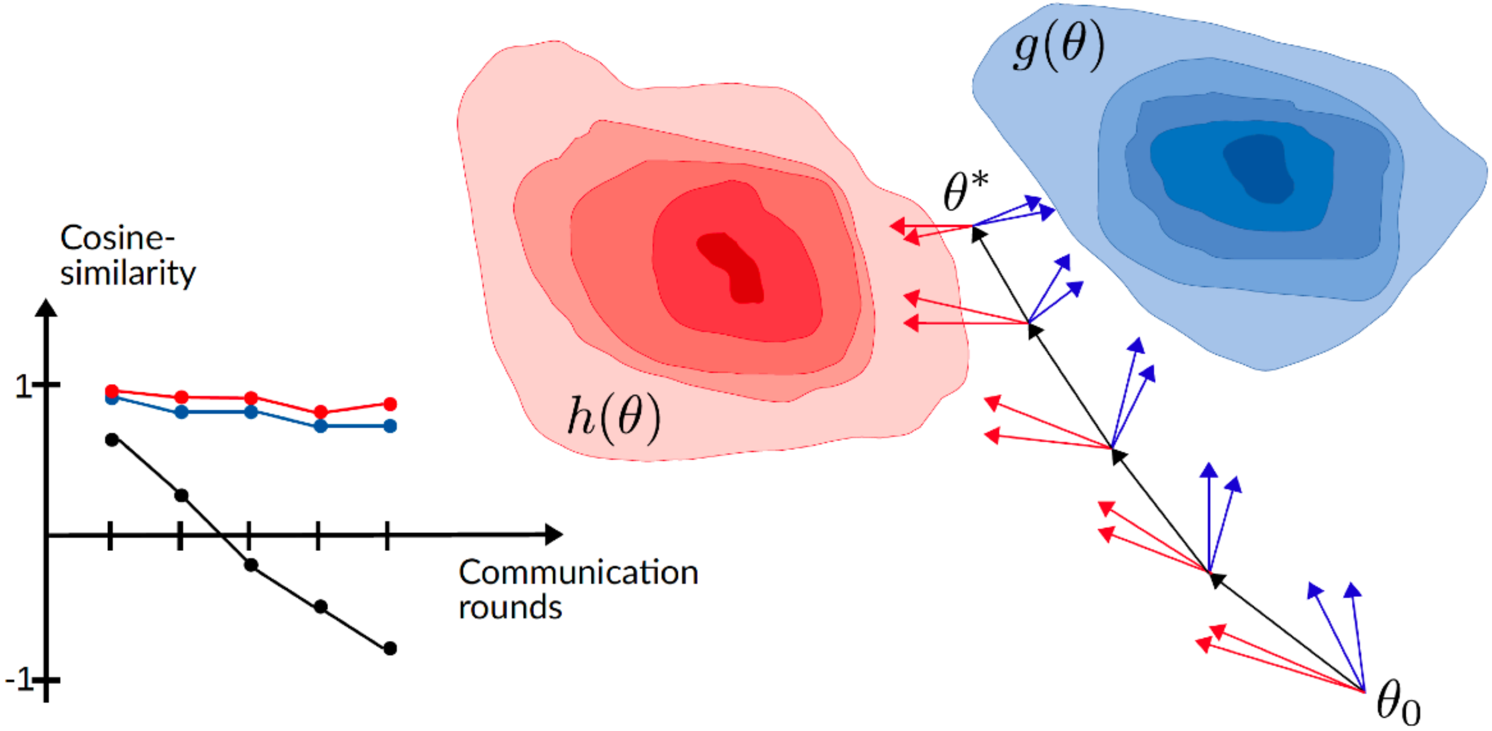
Divergent preferences

$$\nabla g(\theta^*) = -\frac{1-c_1}{c_1}\nabla h(\theta^*)$$

$$\Rightarrow \alpha_{i,j} := \cos(\nabla r_i(\theta^*), \nabla r_j(\theta^*)) = \frac{\langle \nabla r_i(\theta^*), \nabla r_j(\theta^*)\rangle}{\|\nabla r_i(\theta^*)\|\|\nabla r_j(\theta^*)\|}$$

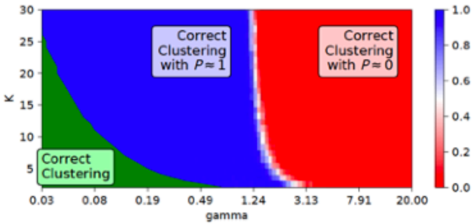
$$= \begin{cases} 1 & I(i) = I(j) \\ -1 & I(i) \neq I(j) \end{cases}$$

Cosine Similarity between Gradients



Separation Theorem

Selection criterion: $c_1, c_2 \leftarrow \arg \min_{c_1 \cup c_2 = c} \left(\max_{i \in c_1, j \in c_2} \alpha_{i,j} \right)$



Separation Theorem:

Let $D_i \sim \varphi_{I(i)}$

$$r_i(\theta) := \frac{1}{|D_i|} \sum_{(x,y) \in D_i} l(f_\theta(x), y)$$

$$R_i(\theta) := \int l(f_\theta(x), y) d\varphi_{I(i)}(x, y)$$

$$F(\theta) := \sum_{i=1}^{m} \frac{|D_i|}{|D|} r_i(\theta)$$

and θ^* s.t. $\nabla_\theta F(\theta^*) = 0$

Then the proposed mechanism will correctly separate the clients if

$$\max_{i=1,..,m} \gamma_i < \sin\left(\frac{\pi}{4(k-1)}\right)$$

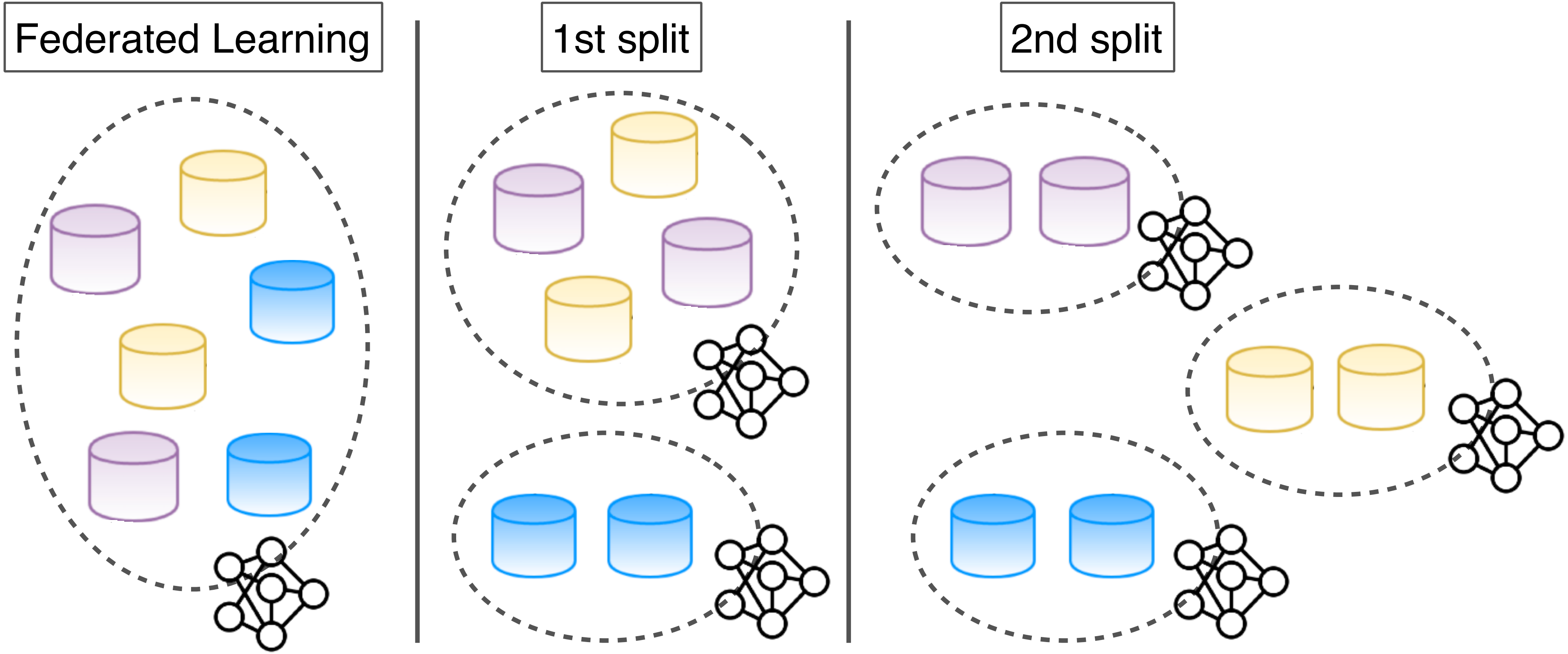
with

$$\gamma_i := \frac{\|\nabla_\theta R_{I(i)}(\theta^*) - \nabla_\theta r_i(\theta^*)\|}{\|\nabla_\theta R_{I(i)}(\theta^*)\|}$$

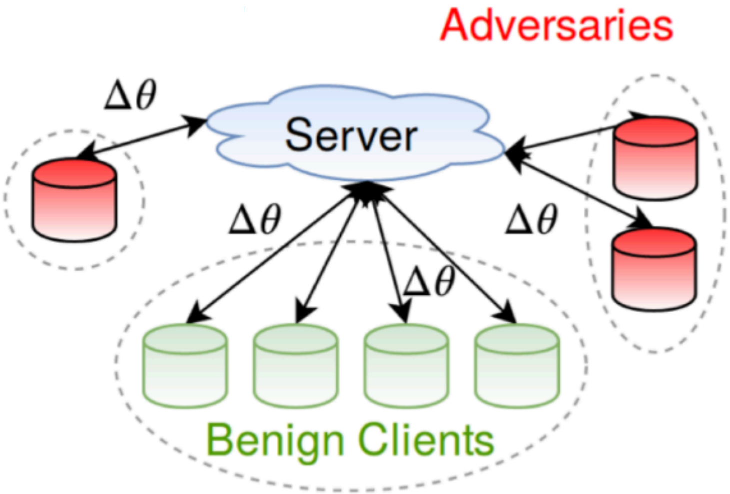
and k being the number of clusters.

[Sattler et al. 2020]

Clustered Federated Learning



Clustered Federated Learning



		Byzantine	Noisy	Label-Flip	Clean
MNIST	FL	9.8%	96.9%	91.3%	**97.5**
	CFL (ours)	**93.19%**	**97.4%**	**97.4%**	97.4%
Fashion-MNIST	FL	9.6%	77.12%	60.6%	79.9
	CFL (ours)	**78.0%**	**79.7%**	**79.7%**	**80.2**
CIFAR	FL	10.0%	70.4%	40.1	**76.0**
	CFL (ours)	**61.7%**	**74.6%**	**74.7%**	75.3%

Byzantine Setting: A subset of clients behaves unpredictably or tries to disturb the joint training effort in an directed or undirected way

[Sattler et al. 2020]

New MPEG Standard

MPEG NNC Standard

Standard on "Compression of Neural Networks for Multimedia Content Description and Analysis"

MPEG NNC Standard

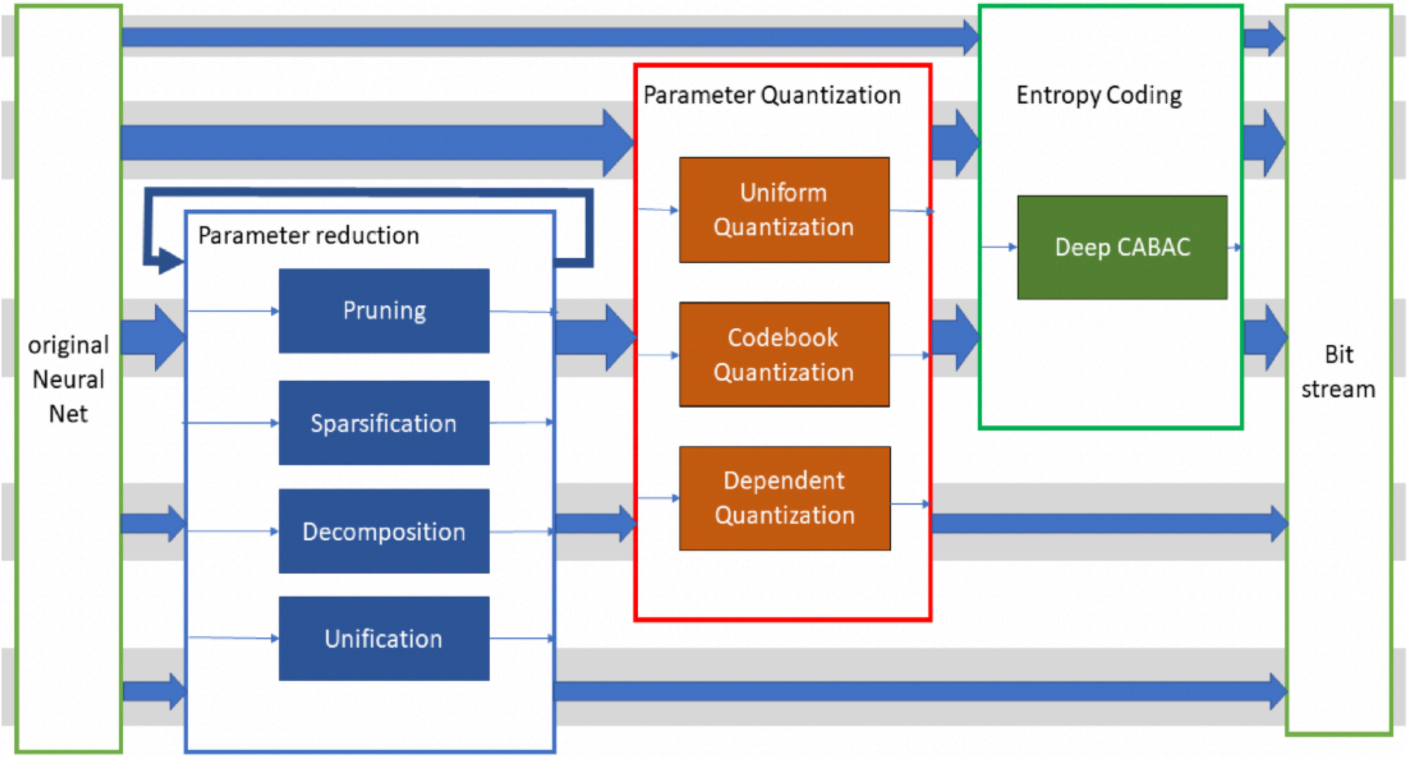
ISO/IEC 15938-17 (MPEG-7 Part 17) – NNC (Compression of neural networks for multimedia content description and analysis)

NNC is developed as a toolbox

- Inclusion in external neural network formats and frameworks, such as PyTorch, TensorFlow, ONNX or NNEF
- Independent coding method of neural networks (compression of weights, biases, etc. per parameter tensor plus inclusion of format-specific topology)

Compression efficiency of 95% without degrading classification quality, e.g. top-1.

MPEG NNC Standard



MPEG: Incremental Compression of NN

Call for Proposals on ICNN issued

Responses by April 2021

Aim: Technology for the compressed, interpretable and interoperable representation of updates of trained neural networks

Target: Use cases on **federated learning** with solution categories on:

- Network updates after refining/adding more training data, e.g. in federated learning
- Network updates after transfer learning/adapting to specific data (with and without network structure changes
- Network updates with higher precision/less compression

Conclusion

Federated Learning is a powerful training schemes (e.g., privacy-preserving, distributed data, ideal for communications applications).

Clustered Federated Learning allows to apply FL in realistic scenarios, where clients have diverging preferences.

Many challenges still exist (complexity, robustness to adversaries etc.)

CFL provides robustness in Byzantine settings, where standard FL fails.

New MPEG standard for FL / NN update compression.

References

Federated Learning

- L Witt, U Zafar, K Shen, F Sattler, D Li, W Samek. Reward-Based 1-bit Compressed Federated Distillation on Blockchain
  arXiv:2106.14265, 2021
  [bibtex] [preprint]
- F Sattler, T Korjakow, R Rischke, W Samek. FedAUX: Leveraging Unlabeled Auxiliary Data in Federated Learning
  IEEE Transactions on Neural Networks and Learning Systems, 2021
  [bibtex] [preprint]
- F Sattler, A Marban, R Rischke, W Samek. CFD: Communication-Efficient Federated Distillation via Soft-Label Quantization and Delta Coding
  IEEE Transactions on Network Science and Engineering, 2021
  [bibtex] [preprint]
- F Sattler, T Wiegand, W Samek. Trends and Advancements in Deep Neural Network Communication
  ITU Journal: ICT Discoveries, 3(1), 2020
  [bibtex] [preprint]
- F Sattler, KR Müller, W Samek. Clustered Federated Learning: Model-Agnostic Distributed Multi-Task Optimization under Privacy Constraints
  IEEE Transactions on Neural Networks and Learning Systems, 32(8):3710-3722, 2021
  [bibtex] [preprint] [supplement]

References

- F Sattler, S Wiedemann, KR Müller, W Samek. Robust and Communication-Efficient Federated Learning from Non-IID Data
  IEEE Transactions on Neural Networks and Learning Systems, 31(9):3400-3413, 2020
  [bibtex] [preprint]
- D Neumann, F Sattler, H Kirchhoffer, S Wiedemann, K Müller, H Schwarz, T Wiegand, D Marpe, and W Samek. DeepCABAC: Plug&Play Compression of Neural Network Weights and Weight Updates
  Proceedings of the IEEE International Conference on Image Processing (ICIP), 21-25, 2020
  [bibtex] [preprint]
- F Sattler, KR Müller, W Samek. Clustered Federated Learning
  Proceedings of the NeurIPS'19 Workshop on Federated Learning for Data Privacy and Confidentiality, 1-5, 2019
  [bibtex] [preprint]
- F Sattler, KR Müller, T Wiegand, W Samek. On the Byzantine Robustness of Clustered Federated Learning
  Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 8861-8865, 2020
  [bibtex] [preprint]
- F Sattler, S Wiedemann, KR Müller, W Samek. Sparse Binary Compression: Towards Distributed Deep Learning with minimal Communication
  Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), 1-8, 2019
  [bibtex] [preprint]

References

NN Compression

- H Kirchhoffer, P Haase, W Samek, K Müller, H Rezazadegan-Tavakoli, F Cricri, E Aksu, MM Hannuksela, W Jiang, W Wang, S Liu, S Jain, S Hamidi-Rad, F Racapé, W Bailer. Overview of the Neural Network Compression and Representation (NNR) Standard
IEEE Transactions on Circuits and Systems for Video Technology, 2021
[bibtex] [preprint]
- P Haase, D Becking, H Kirchhoffer, K Müller, H Schwarz, W Samek, D Marpe, T Wiegand. Encoder Optimizations for the NNR Standard on Neural Network Compression
Proceedings of the IEEE International Conference on Image Processing (ICIP), 2021
[bibtex] [preprint]
- S Wiedemann, S Shivapakashy, P Wiedemanny, D Becking, W Samek, F Gerfers, T Wiegand. FantastIC4: A Hardware-Software Co-Design Approach for Efficiently Running 4bit-Compact Multilayer Perceptrons
IEEE Open Journal of Circuits and Systems, 2:407-419, 2021
[bibtex] [preprint]
- S Yeom, P Seegerer, S Lapuschkin, A Binder, S Wiedemann, KR Müller, W Samek. Pruning by Explaining: A Novel Criterion for Deep Neural Network Pruning
Pattern Recognition, 115:107899, 2021
[bibtex] [preprint]
- K Müller, W Samek, D Marpe. Ein internationaler KI-Standard zur Kompression Neuronaler Netze
FKT- Fachzeitschrift für Fernsehen, Film und Elektronische Medien, 33-36, 2021
[bibtex] [preprint]

References

- S Wiedemann, H Kirchhoffer, S Matlage, P Haase, A Marban, T Marinc, D Neumann, T Nguyen, A Osman, H Schwarz, D Marpe, T Wiegand, W Samek. DeepCABAC: A Universal Compression Algorithm for Deep Neural Networks
IEEE Journal of Selected Topics in Signal Processing, 14(4):700-714, 2020
[bibtex] [preprint] [code]
- S Wiedemann, KR Müller, W Samek. Compact and Computationally Efficient Representation of Deep Neural Networks
IEEE Transactions on Neural Networks and Learning Systems, 31(3):772-785, 2020
[bibtex] [preprint]
- S Wiedemann, T Mehari, K Kepp, W Samek. Dithered backprop: A sparse and quantized backpropagation algorithm for more efficient deep neural network training
Proceedings of the CVPR'20 Joint Workshop on Efficient Deep Learning in Computer Vision, 3096-3104, 2020
[bibtex] [preprint]
- A Marban, D Becking, S Wiedemann, W Samek. Learning Sparse & Ternary Neural Networks with Entropy-Constrained Trained Ternarization (EC2T)
Proceedings of the CVPR'20 Joint Workshop on Efficient Deep Learning in Computer Vision, 3105-3113, 2020
[bibtex] [preprint]

- S Wiedemann, H Kirchhoffer, S Matlage, P Haase, A Marban, T Marinc, D Neumann, T Nguyen, A Osman, H Schwarz, D Marpe, T Wiegand, W Samek. DeepCABAC: A Universal Compression Algorithm for Deep Neural Networks
IEEE Journal of Selected Topics in Signal Processing, 14(4):700-714, 2020
[bibtex] [preprint] [code]

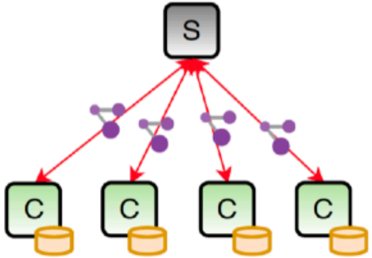
References

- P Haase, H Schwarz, H Kirchhoffer, S Wiedemann, T Marinc, A Marban, K Müller, W Samek, D Marpe, T Wiegand. Dependent Scalar Quantization for Neural Network Compression
  Proceedings of the IEEE International Conference on Image Processing (ICIP), 36-40, 2020
  [bibtex] [preprint]
- S Wiedemann, A Marban, KR Müller, W Samek. Entropy-Constrained Training of Deep Neural Networks
  Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), 1-8, 2019
  [bibtex] [preprint]
- S Wiedemann, KR Müller, W Samek. Compact and Computationally Efficient Representation of Deep Neural Networks
  NIPS Workshop on Compact Deep Neural Network Representation with Industrial Applications (CDNNRIA), 1-8, 2018
  [bibtex] [preprint]

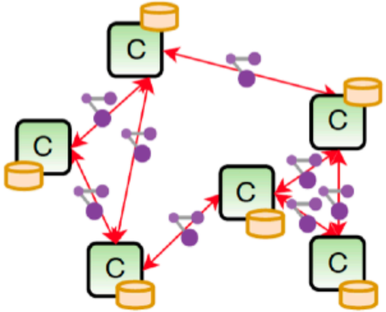
Thank you for your attention!

www.federated-ml.org

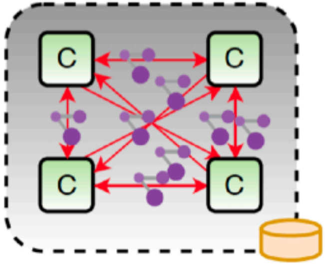
Federated Learning



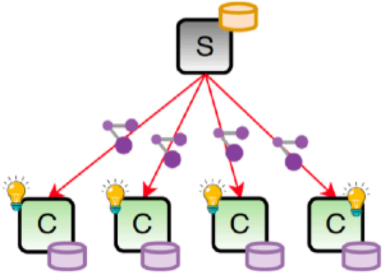
Peer-to-Peer Learning



Distributed Training



On-Device Inference



Papers

Slides

Software