The CCF Advanced Disciplines Lectures
CCFADL第115期 主题: 隐私保护机器学习

Evolutionary Multi-objective Federated Neural Architecture Search

Yaochu Jin, *Distinguished Chair, IEEE Fellow*

Department of Computer Science, University of Surrey

Guildford, UK

yaochu.jin@surrey.ac.uk

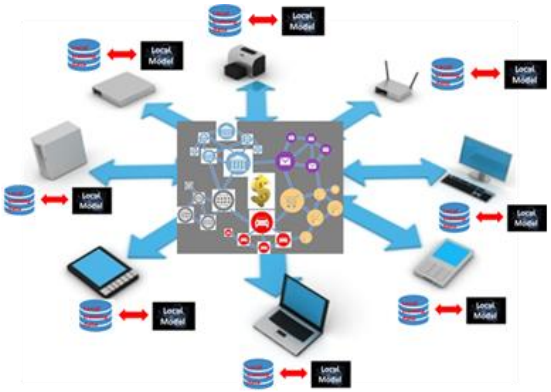
University of Surrey

UNIVERSITY OF SURREY

Research Excellence

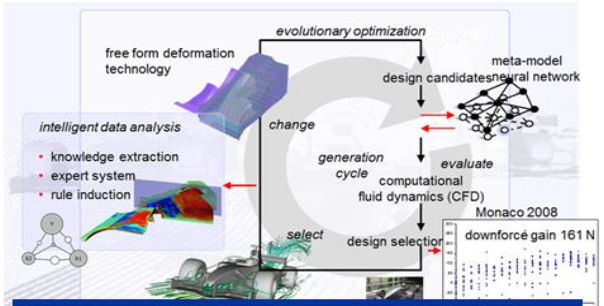
- 5G Innovation Center
- Surrey Space Centre
- Artificial Intelligence

Research Interests



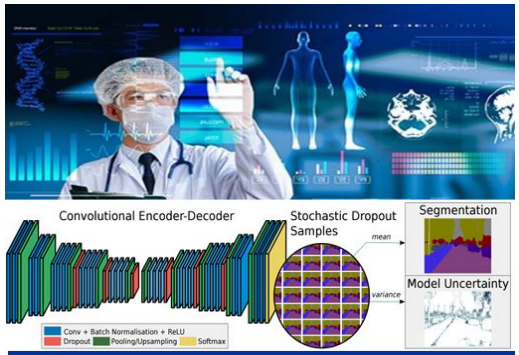
Secure and Privacy-Preserving Machine Learning

HUAWEI



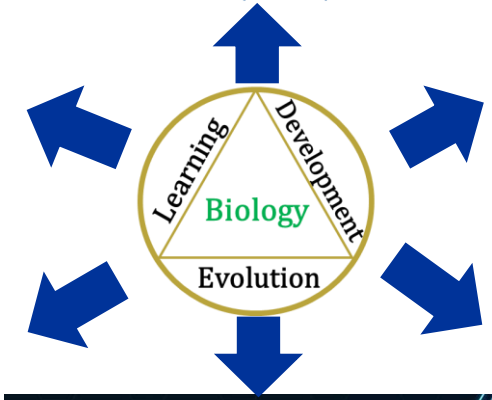
Data-driven Evolutionary Optimization

EPSRC · The Royal Society · Honda The Power of Dreams · Bosch Invented for life · HRI Honda Research Institute EU · Valtra · HR Wallingford Working with water · Airbus · Tekes Finnish Funding Agency for Innovation · QinetiQ · aerooptimal



Statistical Learning and Deep Learning for Healthcare

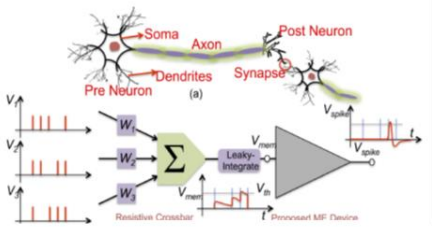
NPL National Physical Laboratory

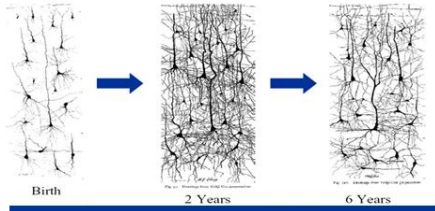


Evolutionary Automated Machine Learning

The Pirbright Institute · ulucu

UNIVERSITY OF SURREY





Learning and Neural Plasticity in Spiking Neural Networks

HP · EPSRC Engineering and Physical Sciences Research Council

Morphogenetic Self-organization of Swarm robots

Seventh Framework Programme · swarm organ

Outline

UNIVERSITY OF SURREY

- Evolutionary multi-objective learning
- Centralized and federated learning
- Evolutionary multi-objective federated structure optimization
- Evolutionary multi-objective federated neural architecture search
- Summary

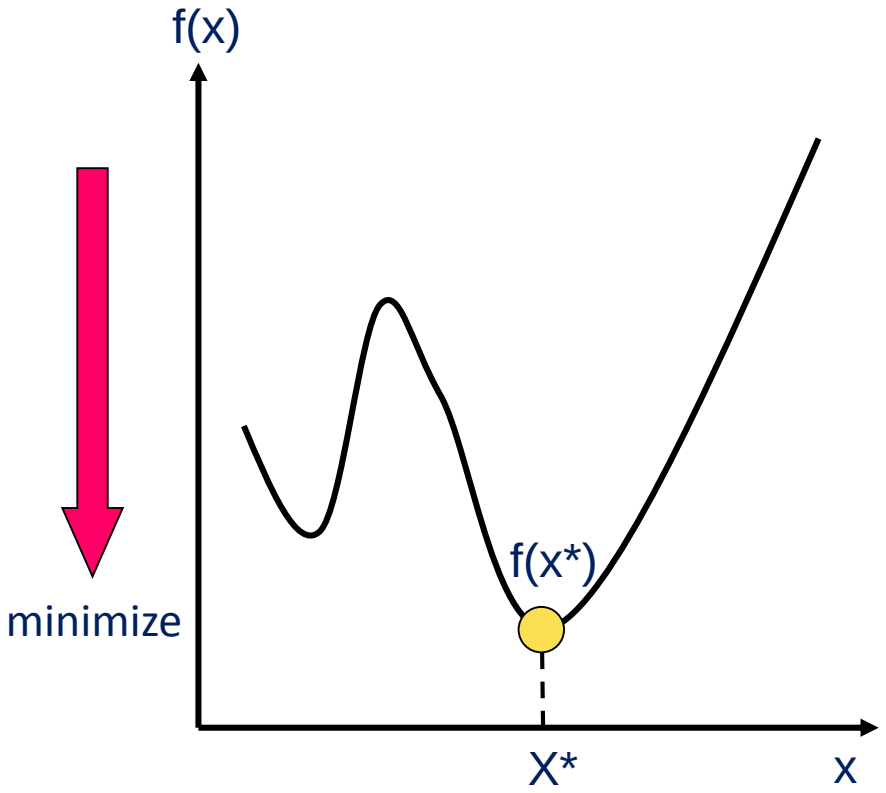
UNIVERSITY OF SURREY

Evolutionary Multi-objective Learning

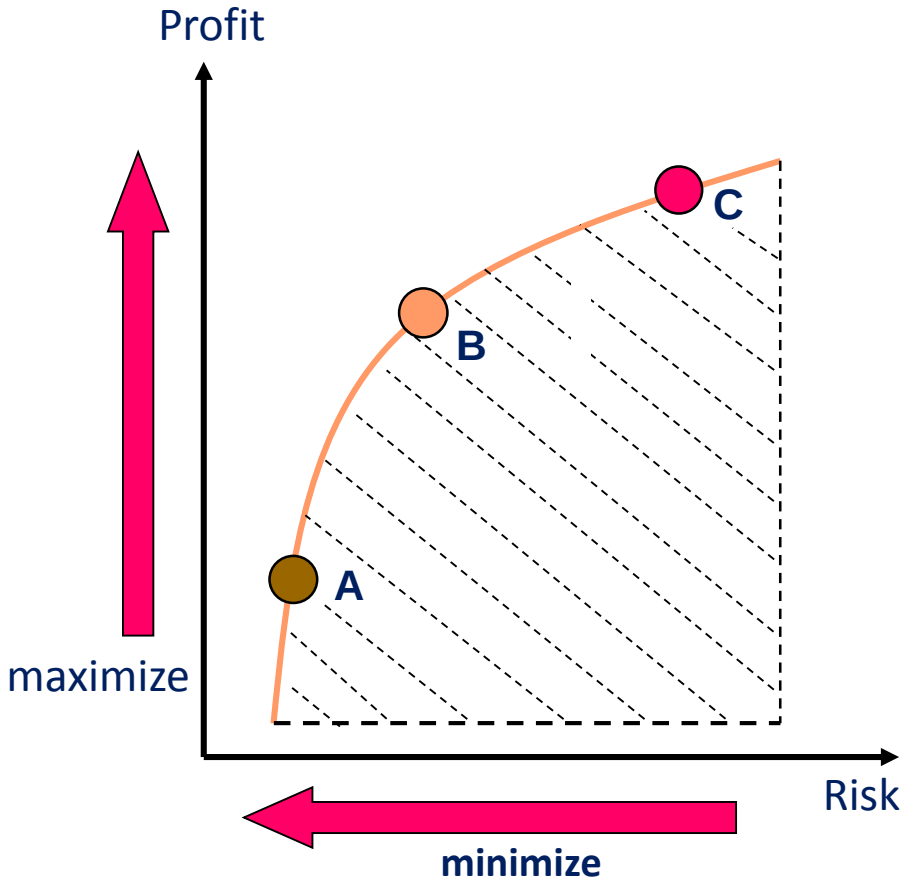
Single and Multi-Objective Optimization

UNIVERSITY OF SURREY

Single-objective optimization (SOO)



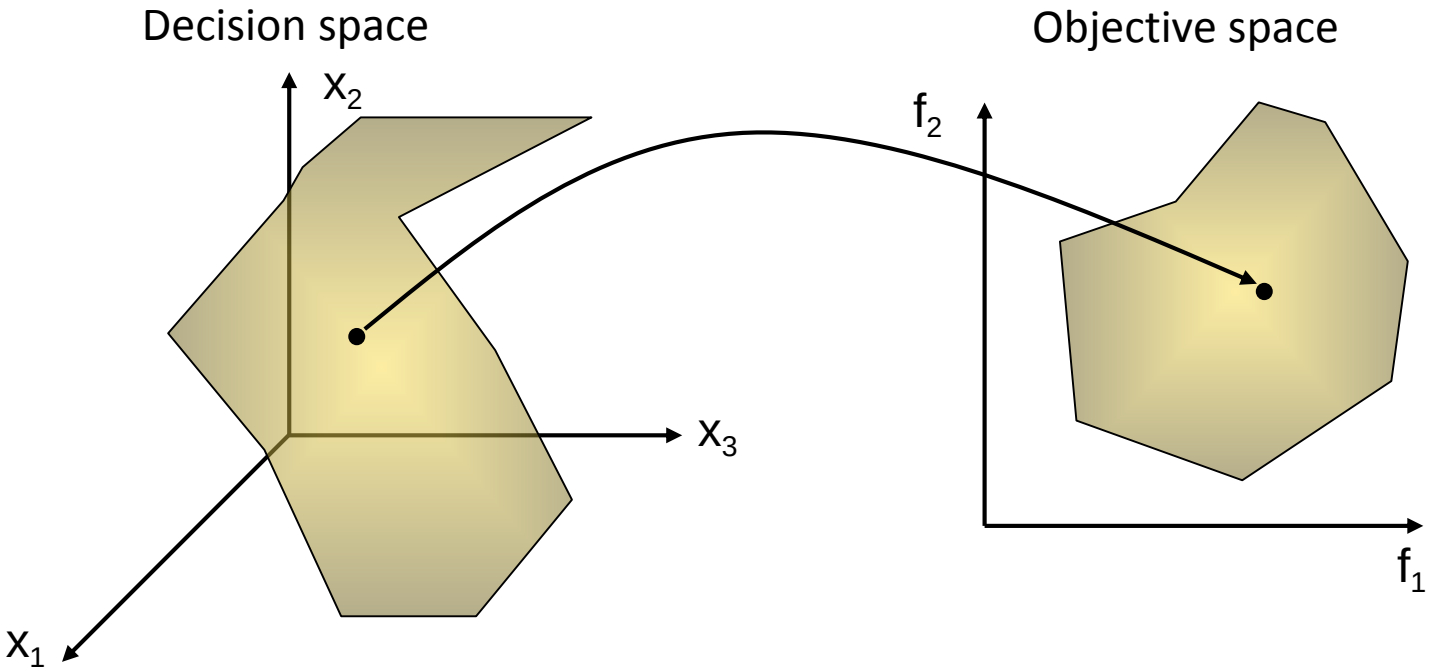
Multi-objective optimization (MOO)



- One single optimal solution can be found for SOO in most cases, whereas a finite or infinite number equally good solutions exist for MOO
- To choose a final solution, user preference is necessary

Mathematical Description of MOO

$$\begin{aligned} &\text{minimize } f_m(\mathbf{X}), && m = 1,2,\ldots, M; \\ &\textit{s.t.} && g_j(\mathbf{X}) \geq 0, && j = 1,2,\ldots, J; \\ & && h_k(\mathbf{X}) = 0, && k = 1,2,\ldots,K; \\ & && x_i^L \leq x_i \leq x_i^U, && i = 1,2,\ldots, n. \end{aligned}$$



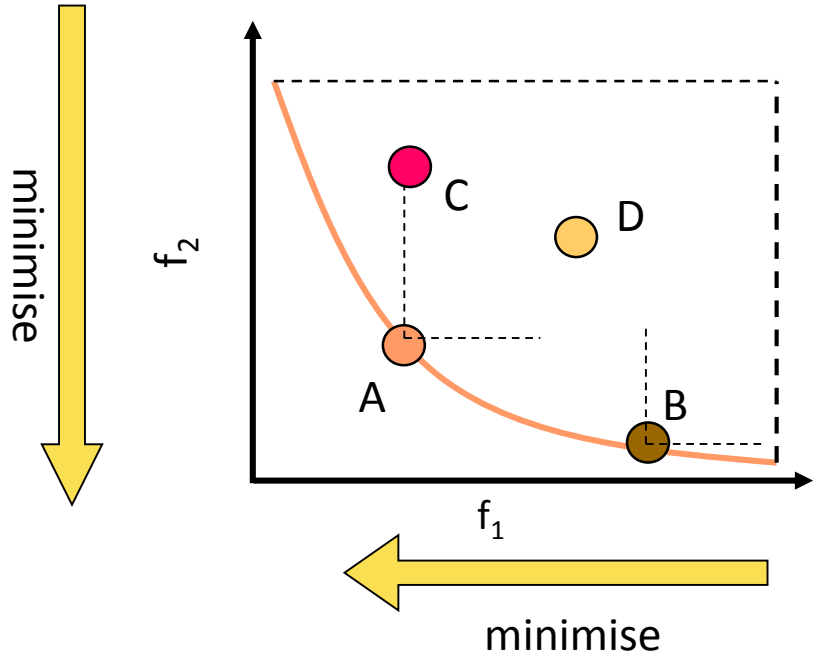
Dominance

- For minimisation problems, solution $\mathbf{X}^{(1)}$ dominates $\mathbf{X}^{(2)}$ if
  - Solution $\mathbf{X}^{(1)}$ is no worse than solution $\mathbf{X}^{(2)}$ in all objectives:

    $$\forall\ m=1,2,\ldots, M,\ f_m(\mathbf{X}^{(1)}) \leq f_m(\mathbf{X}^{(2)}),$$

  - Solution $\mathbf{X}^{(1)}$ is strictly better than $X^{(2)}$ at least in one objective:

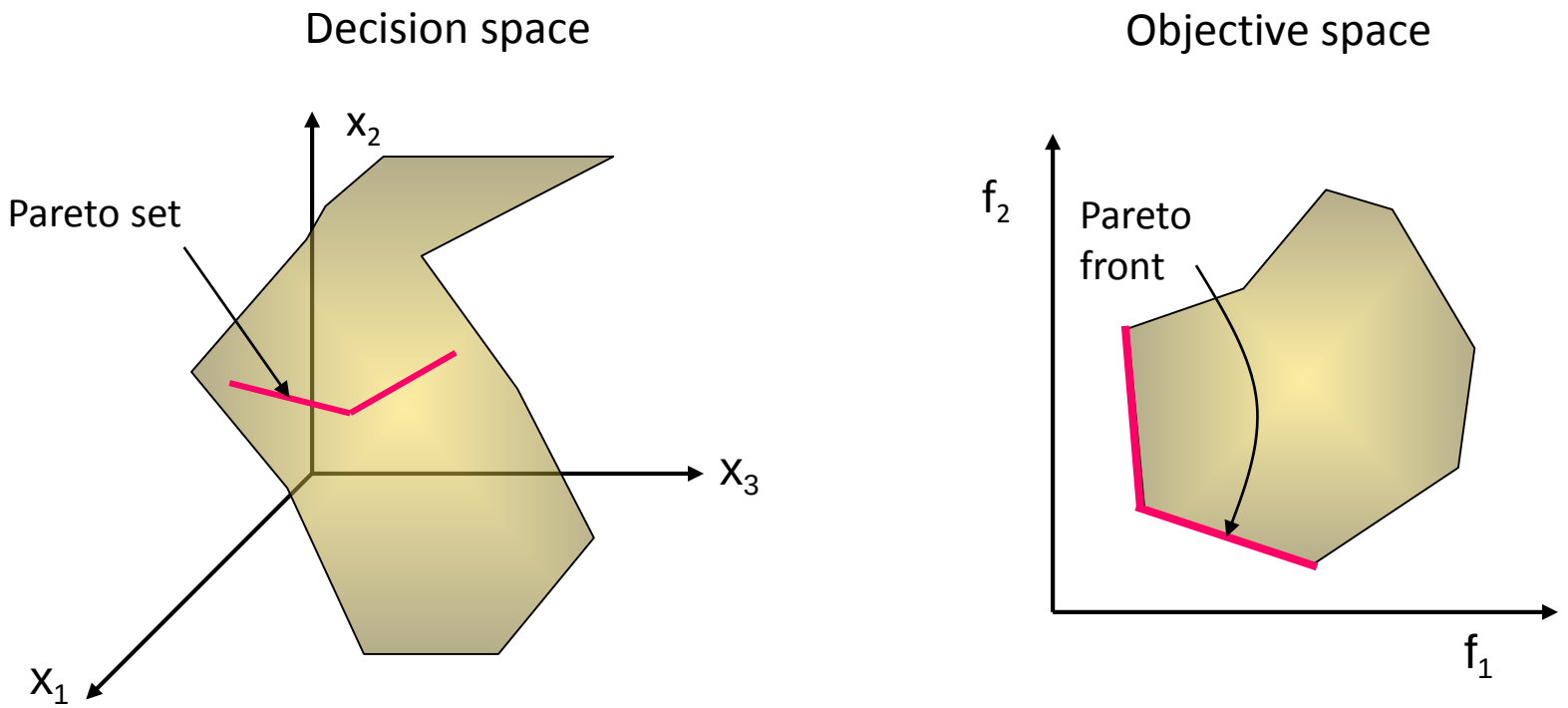
    $$\exists\ m' \in 1,2,\ldots, M,\ f_{m'}(\mathbf{X}^{(1)}) < f_{m'}(\mathbf{X}^{(2)}).$$



- A dominates C and D
- B is not dominated by A

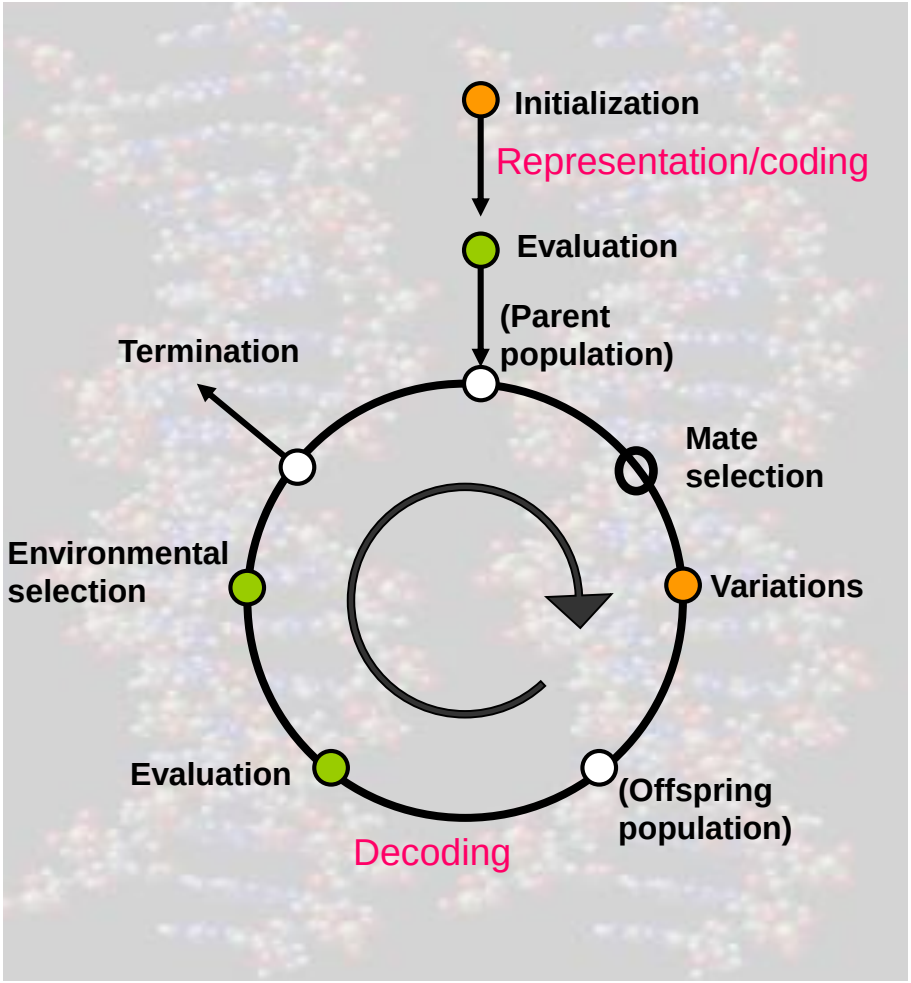
Pareto-Optimal Set and Pareto Front

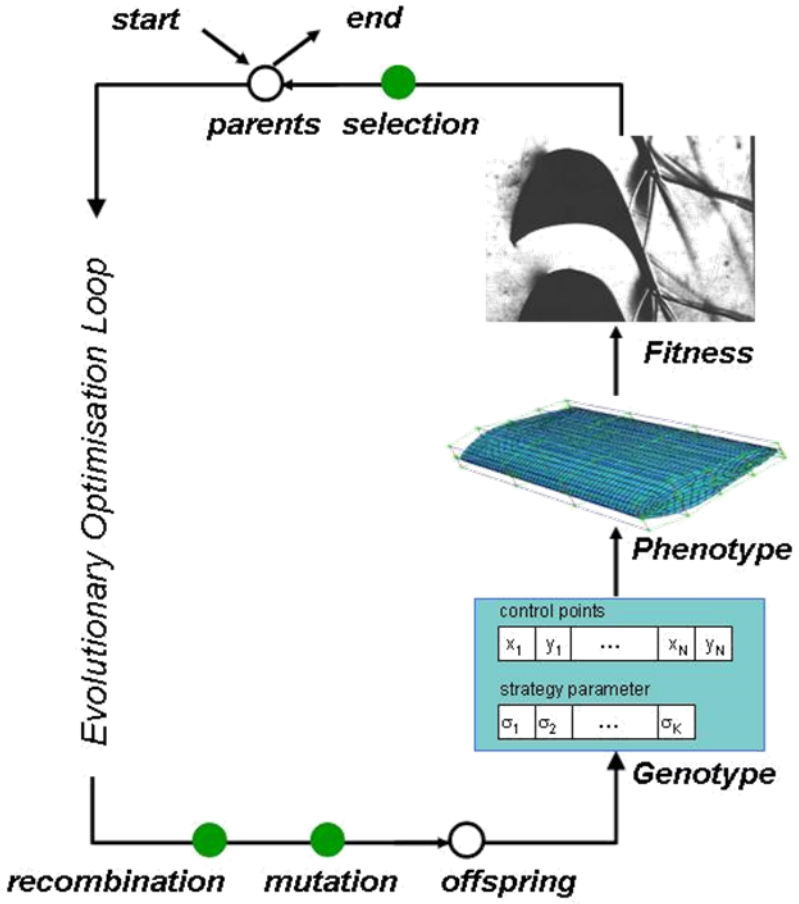
- The set of all the Pareto optimal solutions is called the *Pareto set*. The image of all Pareto optimal solutions in the objective space is termed *Pareto front*.



Evolutionary Algorithms for Optimization

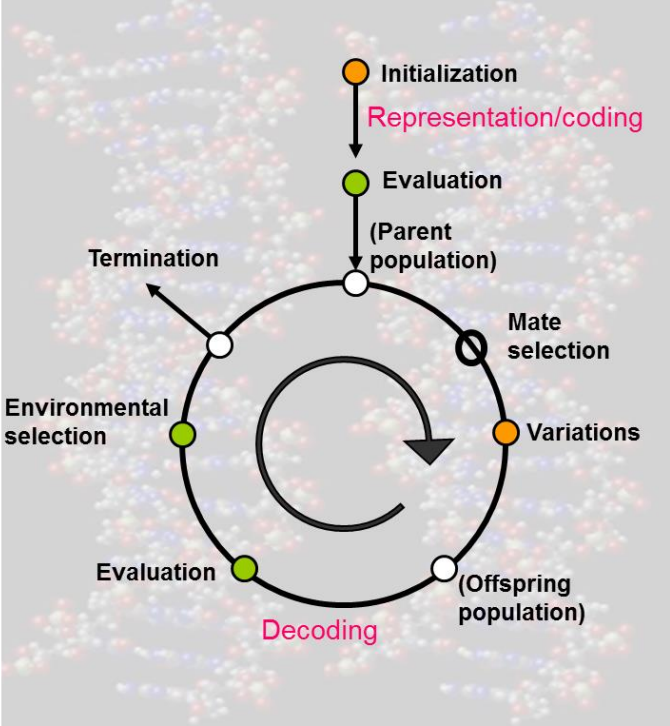
Evolutionary algorithms and other meta-heuristic search methods are a class of population-based, guided stochastic search heuristics inspired from biological evolution and swarm behaviors of social animals





Y. Jin and B. Sendhoff. A systems approach to evolutionary multi-objective structural optimization and beyond. *IEEE Computational Intelligence Magazine*, 4(3):62-76, 2009

Evolutionary Algorithms for Optimization

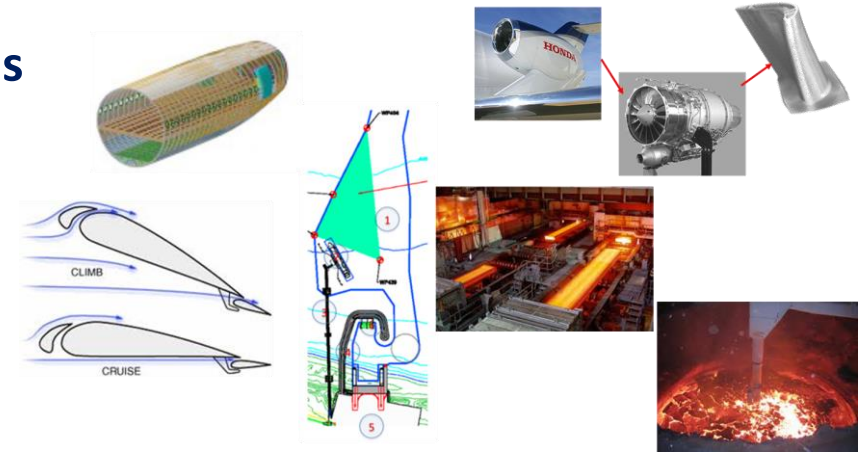


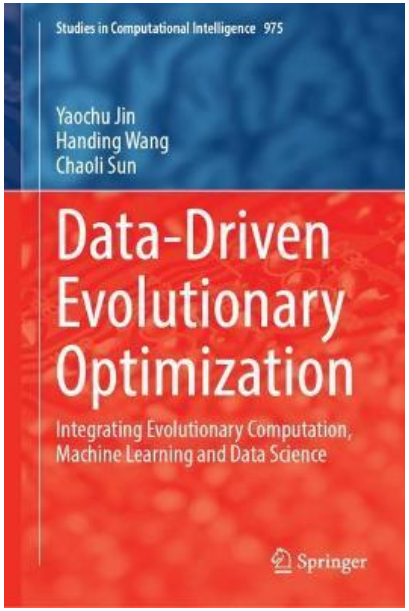
- **Genetic algorithms**
  - Representation: binary / Gray coded
  - Genetic variations: Crossover (main operator), mutation
  - Selection: fitness/rank proportionate selection, tournament selection
- **Genetic programming**
  - Representation: Tree-structure
  - Genetic variations: Crossover and mutation
  - Selection: rank-proportionate
- **Evolution strategies**
  - Representation: Real-coded
  - Genetic variations: Recombination, Gaussian mutations (main)
  - Selection: Elitism ("plus" strategy), non-elitism ("comma" strategy)
- **Other variants**
  - Real-valued genetic algorithms
  - Evolutionary algorithms with hybrid representations
  - Memetic algorithms

Evolutionary Algorithms for Optimization

UNIVERSITY OF SURREY

- **No need for analytical objective functions**
- Strong global search capability
- Less sensitive to uncertainties
- **Well suited for multi-criterion optimisation and mixed-integer optimisation**
- Easy for parallel implementation / distributed computing
- Computationally intensive

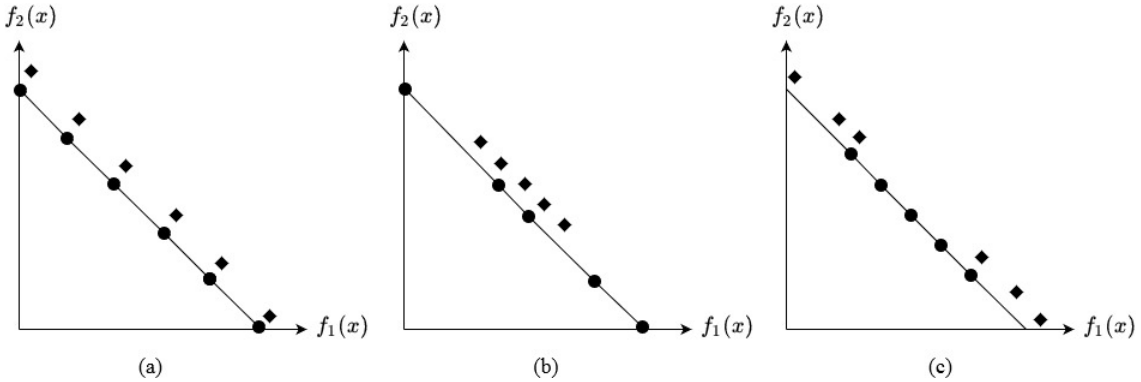




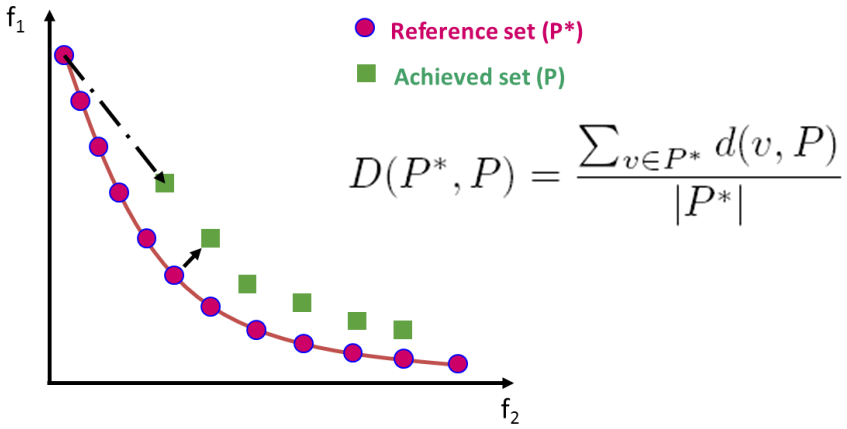
Y. Jin, H. Wang, and C. Sun. Data-Driven Evolutionary Optimization: Integrating Evolutionary Computation, Machine Learning and Data Science. Springer, 2021

Evolutionary Multi-Objective Optimization

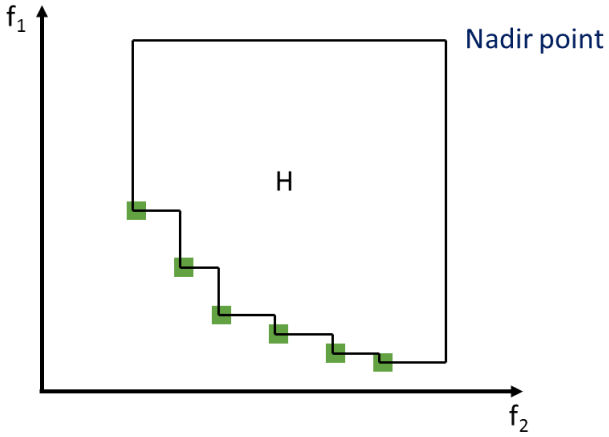
- Basic assumption: To achieve a **representative subset** of the Pareto optimization solutions
  - Accuracy (convergence)
  - Diversity (span, evenness)



- Performance indicators
  - Inverse generational distance (IGD)
  - Hypervolume



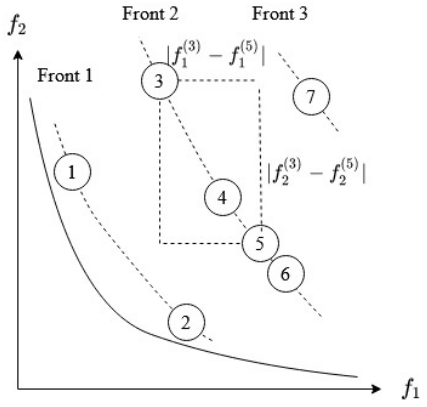
IGD



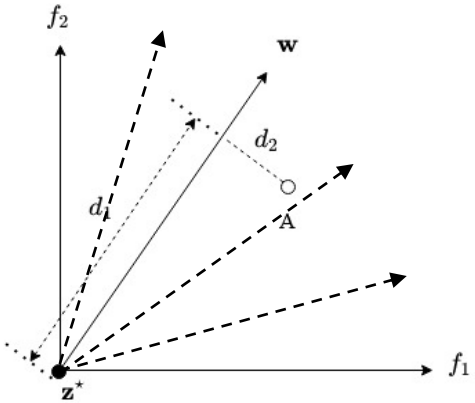
Hypervolume

Evolutionary Multi-Objective Optimization

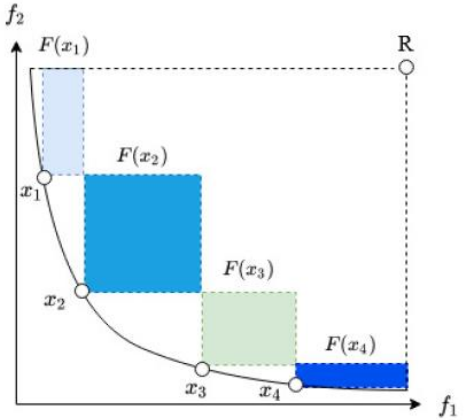
- Main approaches to multi-objective optimization (*a posteriori*)
  - Pareto dominance based approaches
  - Decomposition using weight or reference vectors (cf. a scalarizing function)
  - Performance indicator based approaches



a) Pareto dominance based



b) Decomposition based approaches



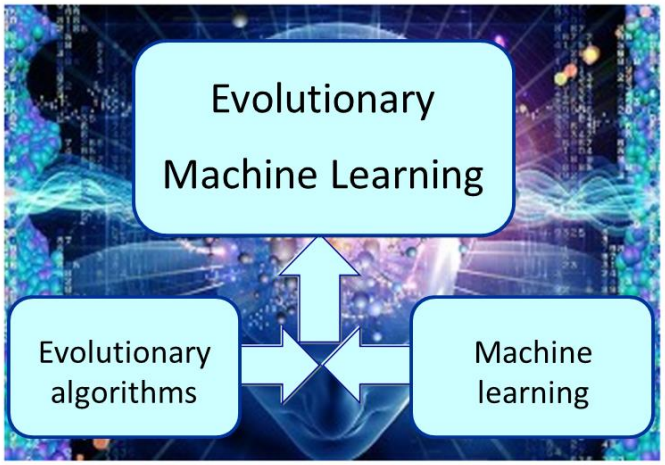
c) Performance indicator

- PlatEMO, a software tool for teaching and research: https://github.com/BIMK/PlatEMO, which contains over **150** open source evolutionary algorithms and **300** benchmark and application problems

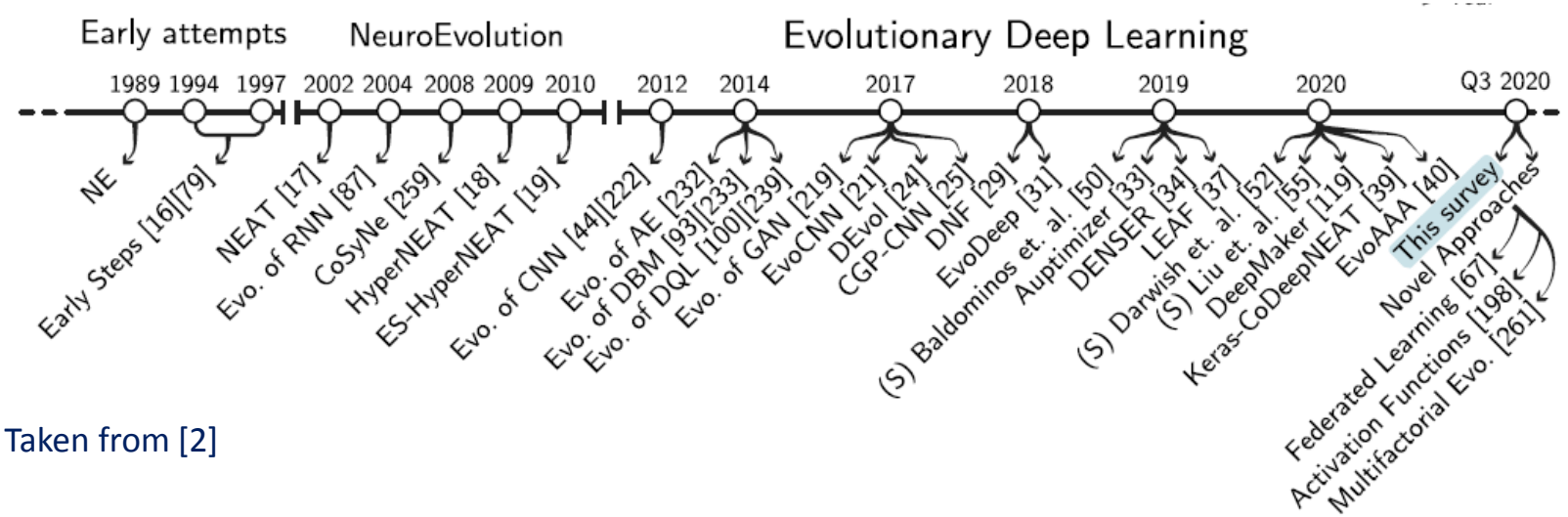
PlatEMO

Y. Tian, R. Cheng, X. Zhang, and Y. Jin. **PlatEMO**: A MATLAB platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine*, 12(4): 73-87, 2017 ("**2019 IEEE CIM Outstanding Paper Award**")

Evolutionary Machine Learning



- Evolutionary learning is
  - able to solve non-convex learning problems
  - good for both (hyper)-parameter and structure optimization
  - good for multi-objective machine learning
  - good for **automated machine learning**

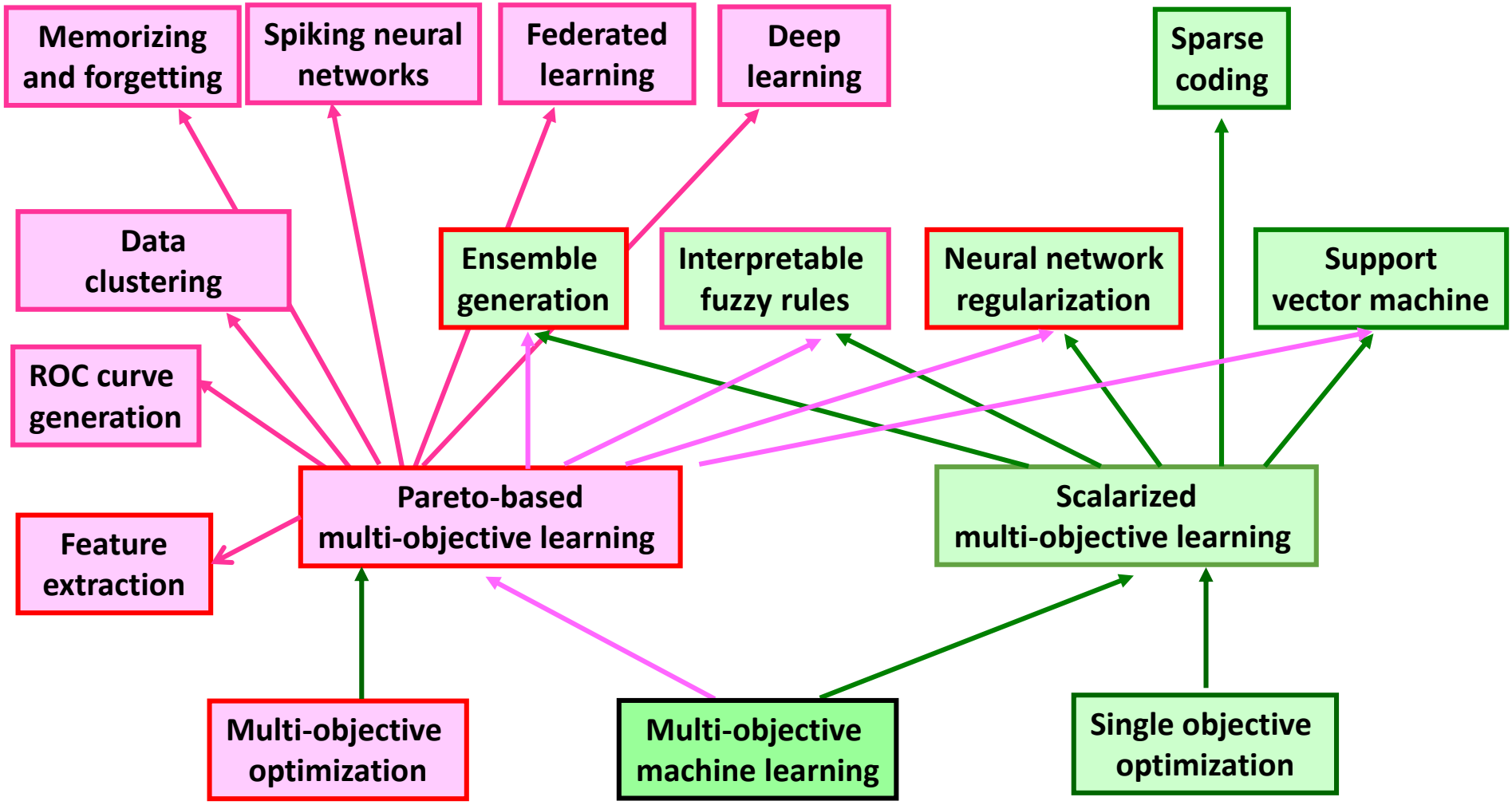


Taken from [2]

[1] X. Yao. Evolving artificial neural networks. *Proceedings of the IEEE,* 87 (9):1423-1447, 1999

[2] A.D. Martinez *et al*. Lights and shadows in Evolutionary Deep Learning: Taxonomy, critical methodological analysis, cases of study, learned lessons, recommendations and challenges. *Information Fusion,* 67:161–194, 2021

Evolutionary Multi-Objective Machine Learning



- Y. Jin and B. Sendhoff. Pareto-based multi-objective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(3):397-415, 2008
- Y. Jin (ed.) Multi-objective machine learning. Springer, 2006

Learning with Regularization

- A complexity term is included in the cost function

$$J = E + \lambda \Omega$$

E -- Error function, Ω -- complexity

λ -- hyper-parameter

  - Need to predefine a proper hyper-parameter

- Gaussian and Laplacian regularizers

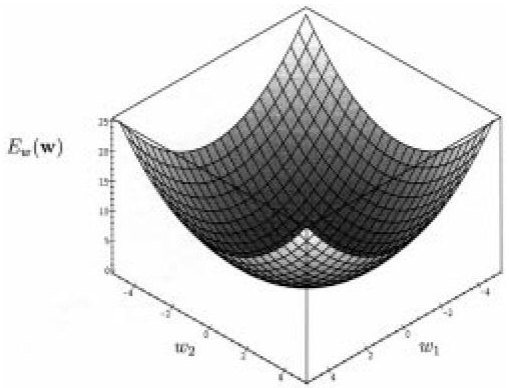
- Gaussian regularizer

$$\Omega = \sum_{i=1}^{M} w_i^2,$$

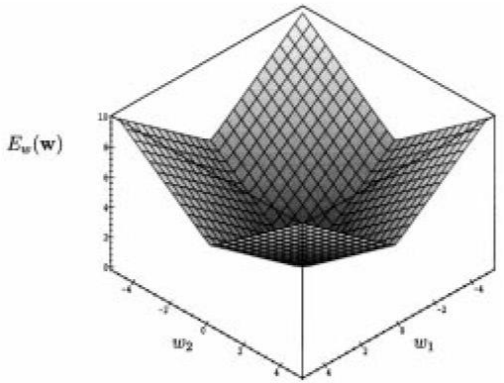
- Laplacian regularizer

$$\Omega = \sum_{i=1}^{M} |w_i|$$

  - Laplacian regularizer is believed to be more effective in reducing complexity



Gaussian



Laplacian

Pareto Approach to Neural Regularization

UNIVERSITY OF SURREY

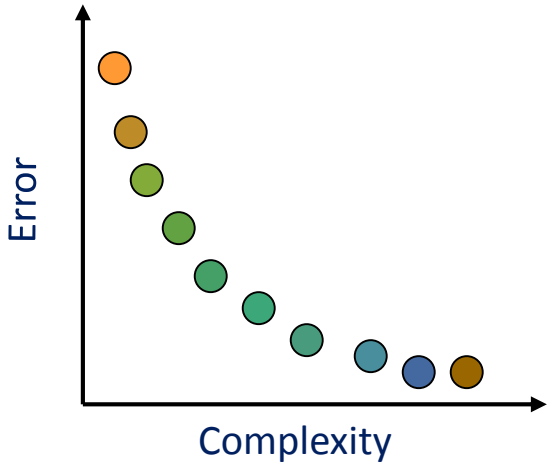
$$\min \{f_1, f_2\}$$

$$f_1 = E;$$

$$f_2 = \Omega.$$

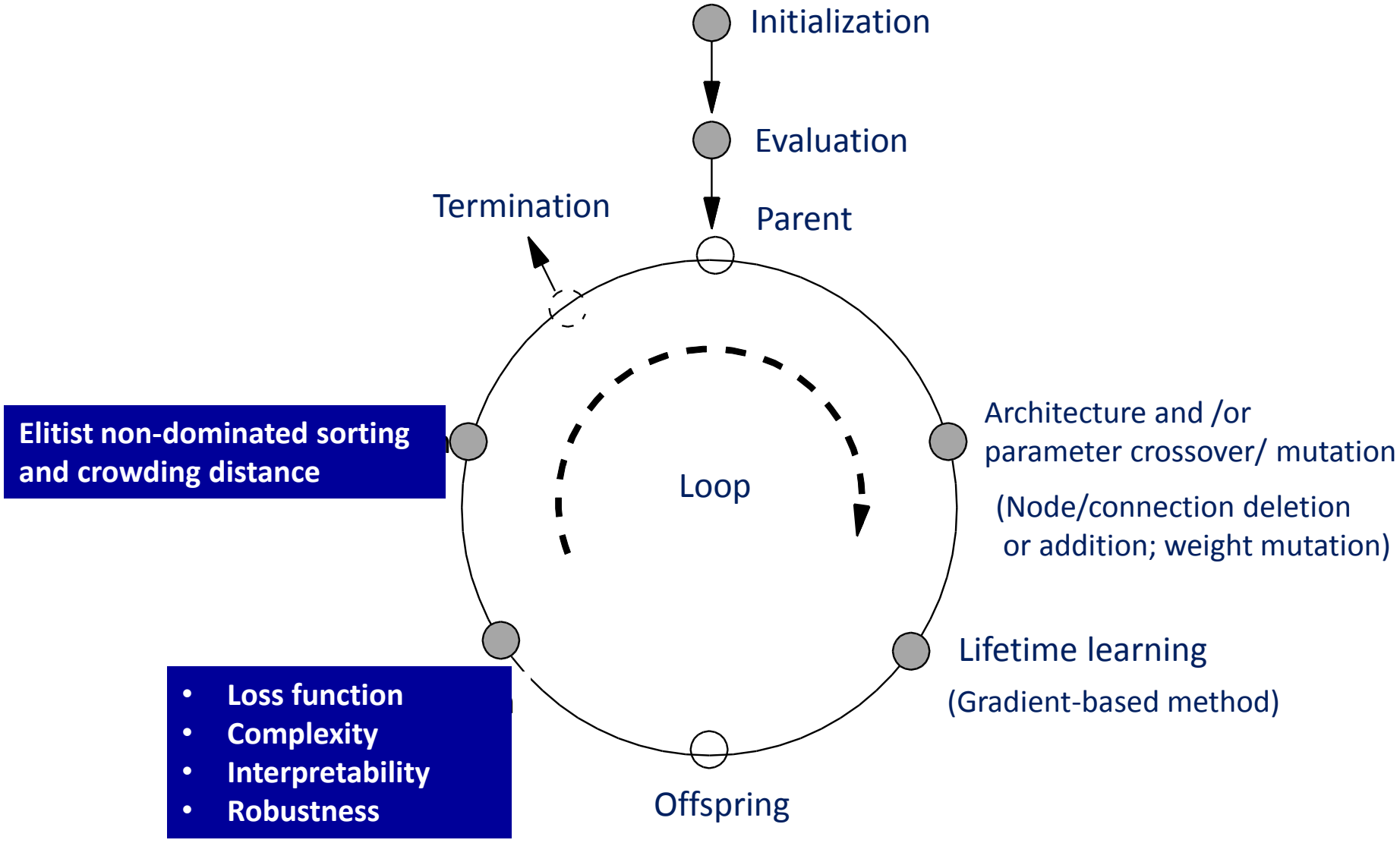
E: approximation error, Ω: complexity

- Gaussian regularizer
- Laplace regularizer
- Number of connections / neurons



- Instead of a single model, multiple models with a spectrum of complexity can be obtained simultaneously

Single- and Multi-objective Evolutionary Learning

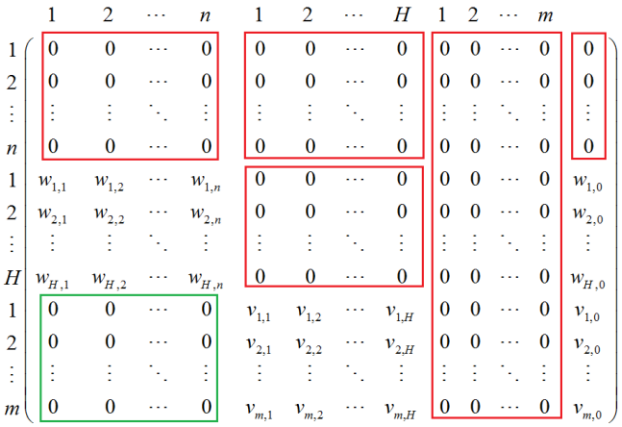


Y. Jin, B. Sendhoff. Pareto-based multi-objective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(3):397-415, 2008

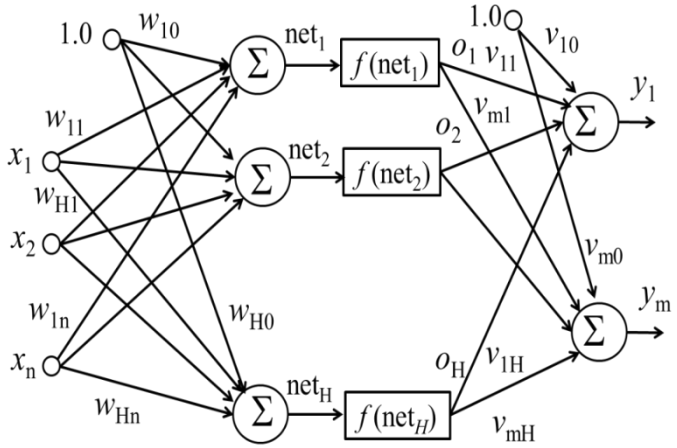
Structure Optimization - Direct Representation

UNIVERSITY OF SURREY

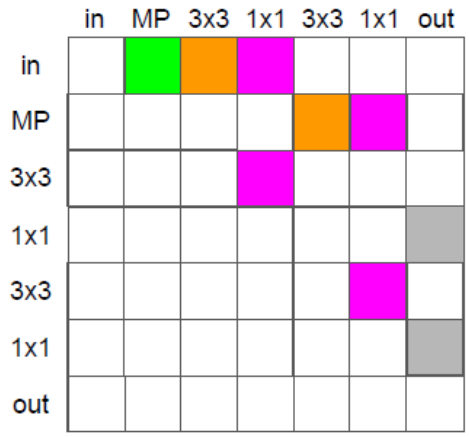
- Direct architecture and weight representation
  - ➢ a connection matrix
  - ➢ a weight matrix
- Poor scalability in particular for deep neural networks

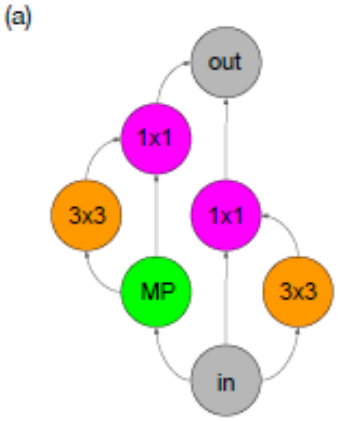


Connection matrix



Network architecture





Analyses of Pareto Front

- By analyzing the “accuracy-complexity” Pareto front, we are able to gain deeper insights into the learning problem
  - Identify Pareto-optimal solutions of low complexity from which interpretable rules can be extracted
  - Identify networks that are able to generalize on unseen data
  - Identify well-performed networks with diverse structures for building ensembles

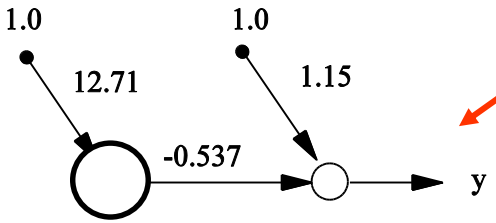
Y. Jin, R. Grunar, and B. Sendhoff. Pareto analysis of evolutionary and learning systems. *Frontiers of Computer Science in China*, 3(1):4-17, 2009

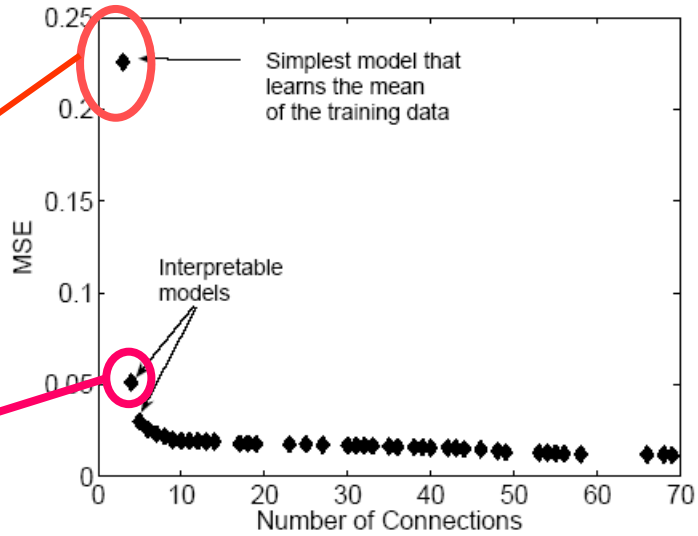
Rule Extraction from Simple NN

- Breast cancer diagnosis (BCD) data, nine attributes, two classes (benign, malignant)

- Simplest Pareto-optimal: No input feature is chosen, 3 connections

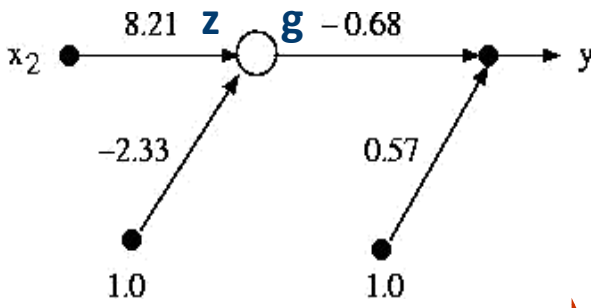
y = 0.65





BCD data

- Pareto-optimal NN1: only 1 input feature (x_2) is chosen, 4 connections



If y < 0.25, then benign
If y > 0.75, then malignant

-0.68g + 0.57 < 0.25 → benign
-0.68g + 0.57 > 0.75 → malignant

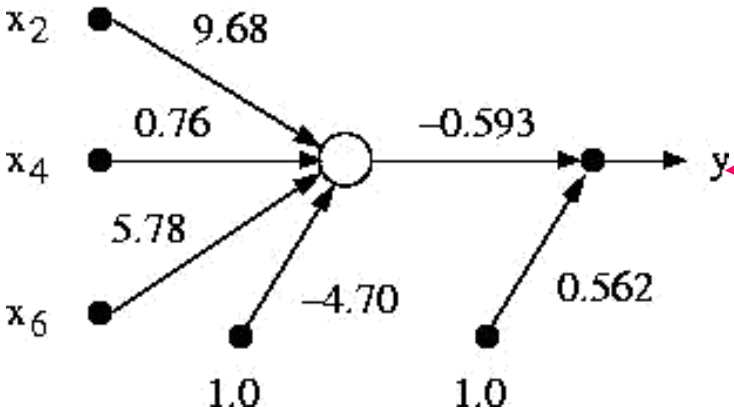
$g = z / (1 + |z|)$
$z = 8.21\ x_2 - 2.33$

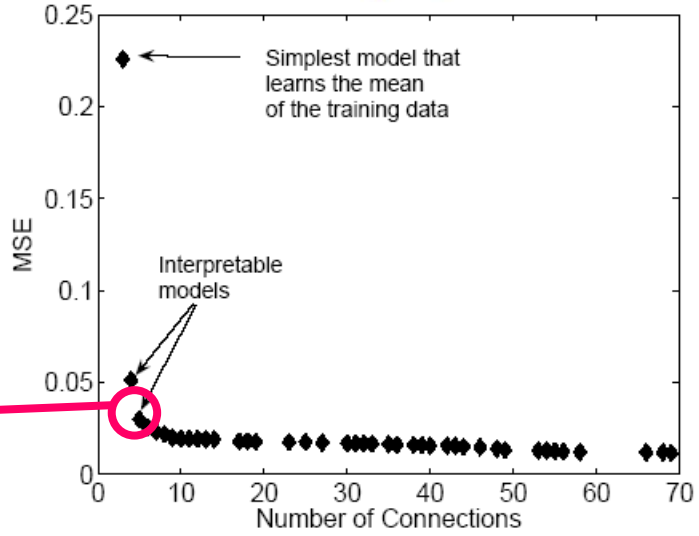
R1: If $x_2 \geq 0.5$, then malignant;
R2: If $x_2 \leq 0.2$, then benign

Rule Extraction from Simple NN

Pareto-optimal NN2:

- 6 connections
- 3 input features (x2, x4 and x6) are chosen





R1: If $x_2 \geq 0.6 \vee x_6 \geq 0.9 \vee x_2 \geq 0.5 \wedge x_6 \geq 0.2 \vee$
$x_2 \geq 0.4 \wedge x_6 \geq 0.4 \vee x_2 \geq 0.3 \wedge x_6 \geq 0.5 \vee$
$x_2 \geq 0.2 \wedge x_6 \geq 0.7$, then malignant;

R2: If $x_2 \leq 0.1 \wedge x_6 \leq 0.4 \vee x_2 \leq 0.2 \wedge$
$x_6 \leq 0.2$, then benign

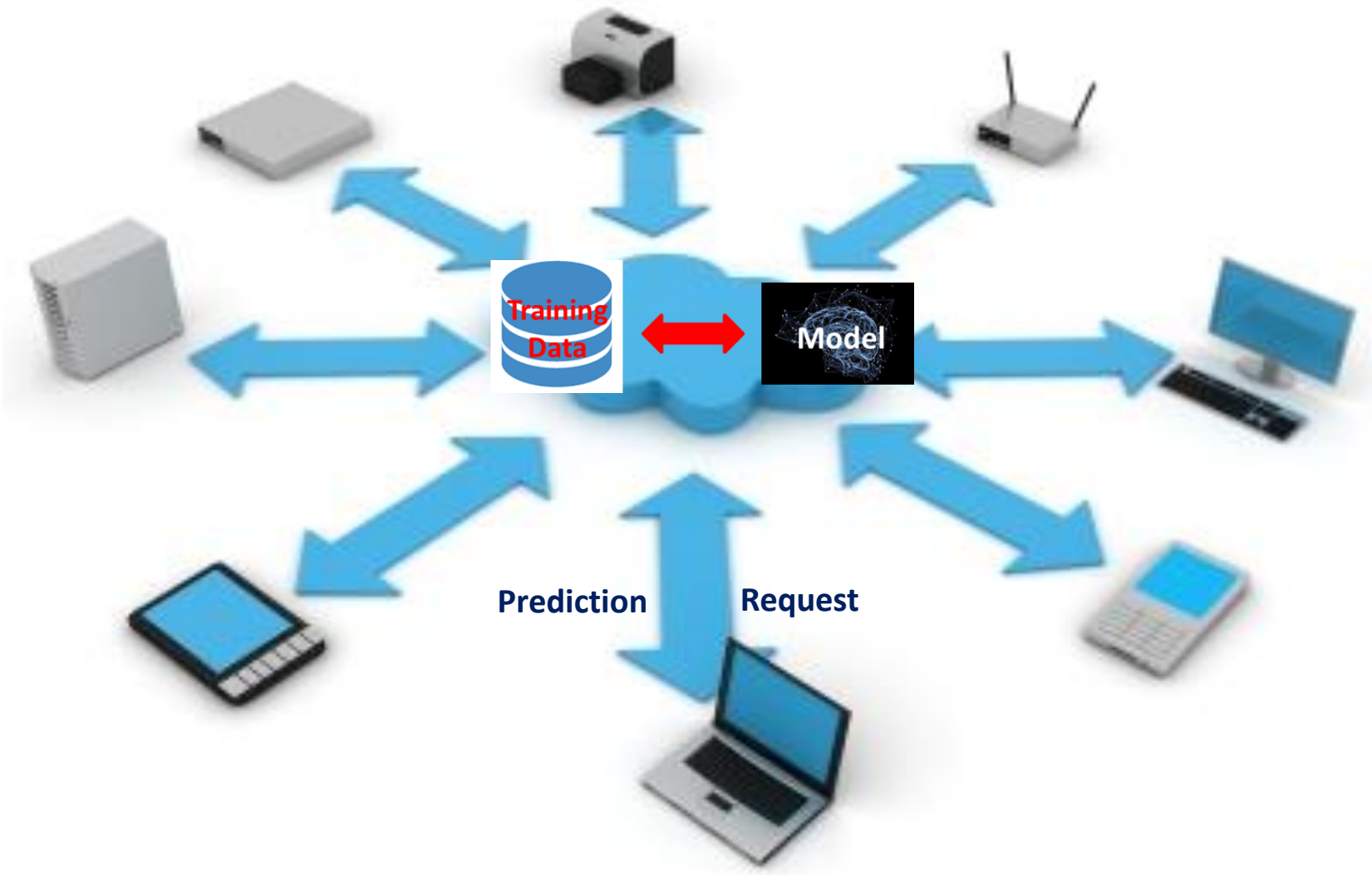
(x_4 is too weak to play any role in the rules)

UNIVERSITY OF SURREY

Privacy-Preserving Machine Learning

Centralized Learning in the Cloud

UNIVERSITY OF SURREY



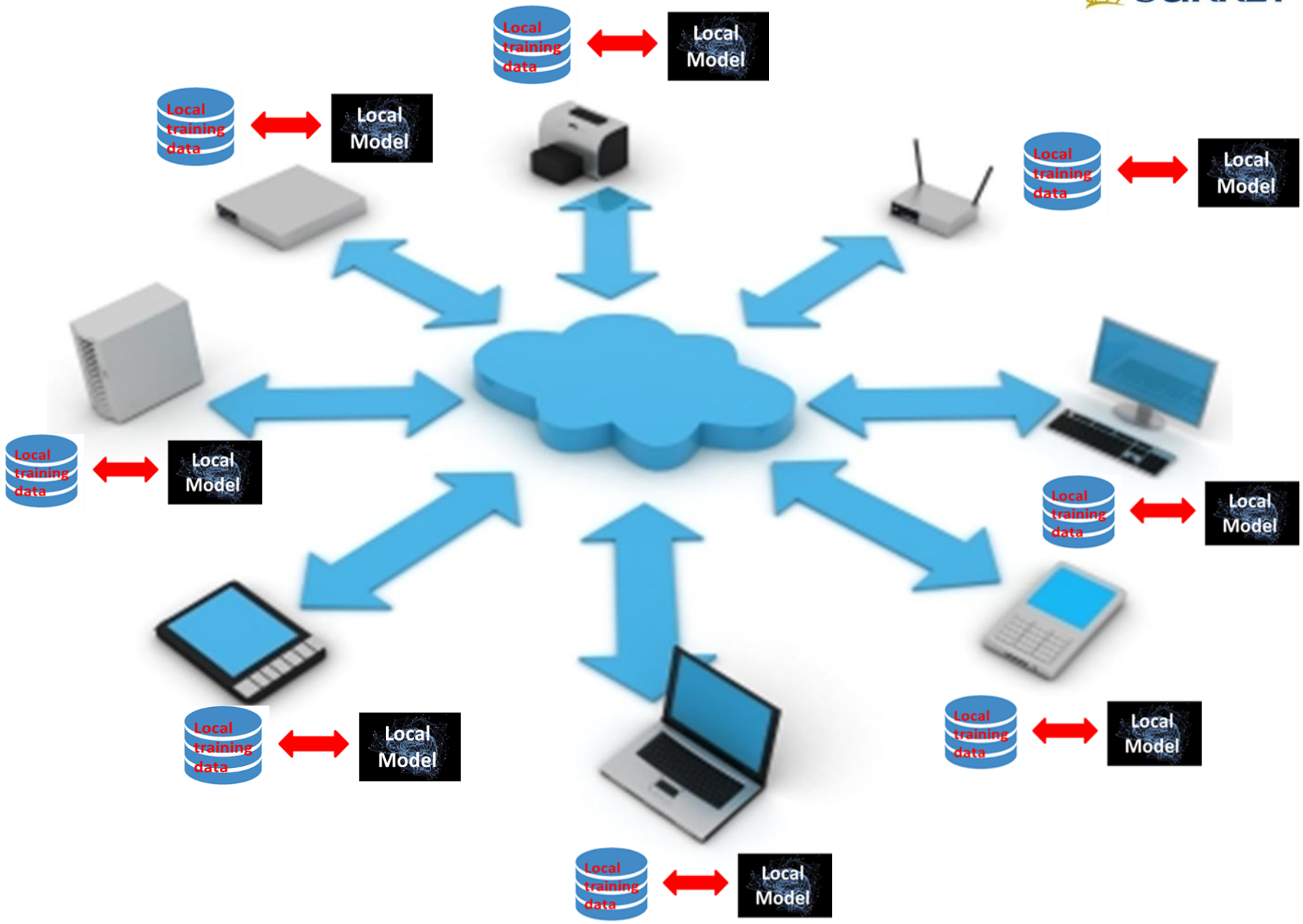
Advantages and Disadvantages

- Advantaged
  - All data are collected and stored in the cloud
  - The machine learning model is also trained in the cloud
  - An accurate model can be achieved

- Disadvantaged
  - Data from the edge devices must be sent to the cloud, causing potential security and privacy issues

Distributed On-Device Learning

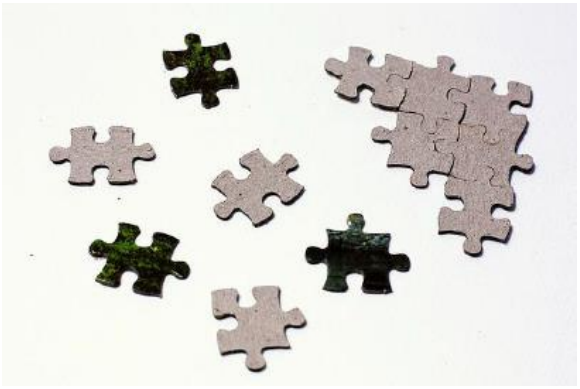
UNIVERSITY OF SURREY



Advantages and Disadvantages

- Advantages
  - Learning occurs on the edge devices
  - No data collection is needed
  - Less security concerns

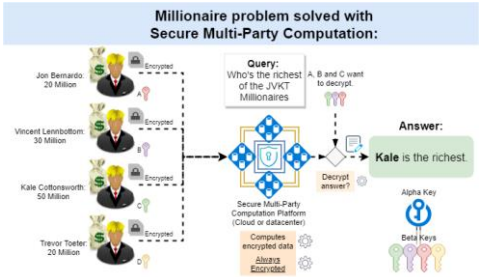
- Disadvantages
  - The amount of data may be **limited and non-iid** (not independent and identically distributed), and thus the model quality may be low
  - Not able to learn from others' experience , and worse if there is no data for a new user



Privacy Preserving Techniques

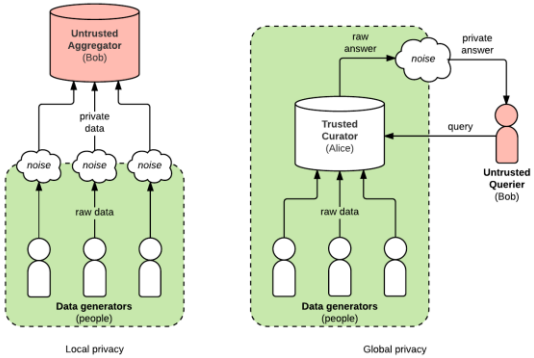
- **Secure multi-party computation**: Protocols for a set of parties to jointly compute a function over their inputs while keeping those inputs private
  - Zero knowledge
  - Partial knowledge

  Computationally inefficient and data needs to be secretly shared



- **Differential privacy:** an algorithm is said to be differentially private if by looking at the output, one cannot tell whether any individual's data was included in the original dataset or not
  - Add noise
  - Make data obscure

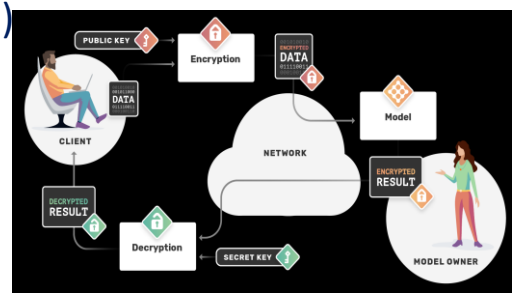
  Tradeoff between privacy and accuracy



- **Homomorphic encryption:**
  - Partially homomorphic encryption (addition or multiplication)

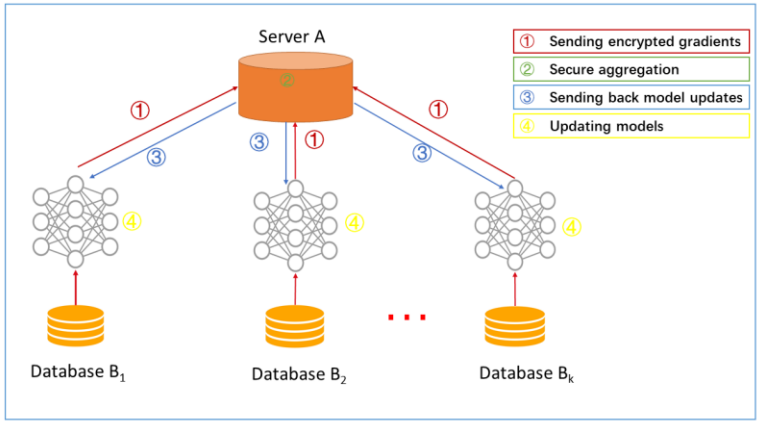
  Additive: E [a] * E [b] = E [a+b]; Multiplication: E [a] * E [b] = E [a*b]; '*' is a predefined function
  - Fully homomorphic encryption (unlimited addition and multiplication)
  - Somewhat homomorphic encryption (limited operations of FHE)

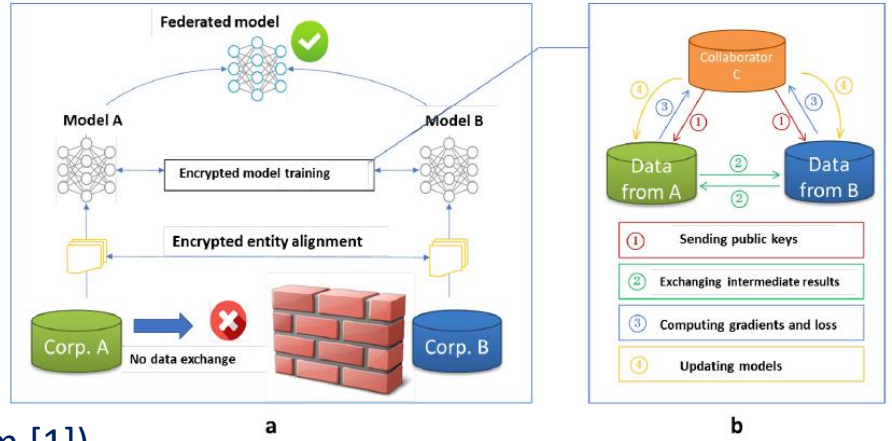
  Limited machine learning models can be applied



Federated Learning

- **Federated learning** is a machine learning setting where the goal is to train a high-quality *centralized model* with training data *distributed* over a large number of clients, each with *unreliable* and relatively *slow* network connections.





(Taken from [1])

DB1

DB2

	ID	X1	X2	X3	ID	X4	X5	Y
DB1	U1	9	80	600	U1	6000	600	No
	U2	4	50	550	U2	5500	500	Yes
	U3	2	35	520	U3	7200	500	Yes
DB2	U4	10	100	600	U4	6000	600	No
	U5	5	75	600	U8	6000	600	No
DB3	U6	5	75	520	U9	4520	500	Yes
	U7	8	80	600	U10	6000	600	No

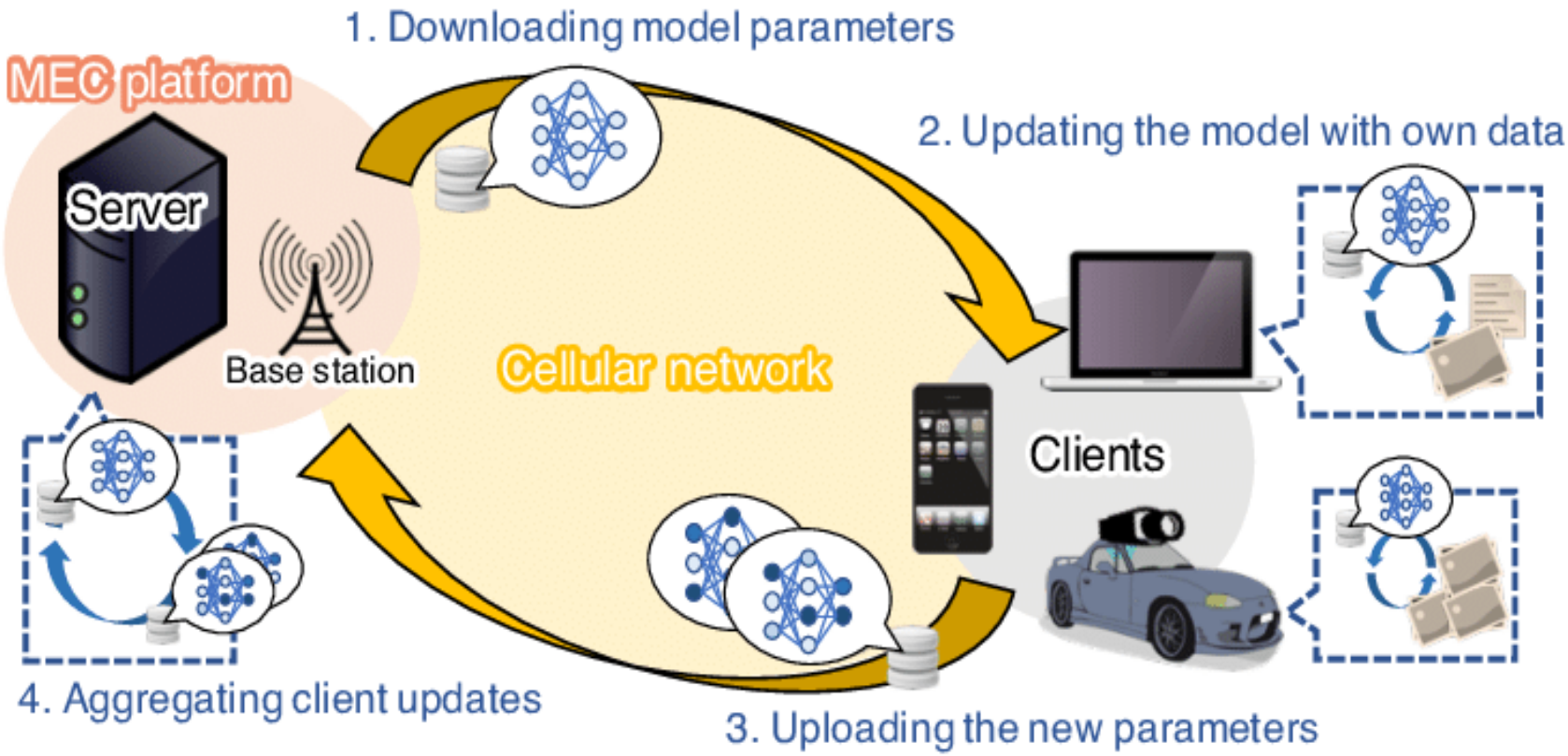
Horizontal federated learning

ID	X1	X2	X3	ID	X4	X5	Y
U1	9	80	600	U1	6000	600	No
U2	4	50	550	U2	5500	500	Yes
U3	2	35	520	U3	7200	500	Yes
U4	10	100	600	U4	6000	600	No
U5	5	75	600	U8	6000	600	No
U6	5	75	520	U9	4520	500	Yes
U7	8	80	600	U10	6000	600	No

Vertical federated learning

[1] Yang et al., Federated Machine Learning: Concept and Applications. https://arxiv.org/abs/1902.04885

Federated Learning



Main Challenges in Federated Learning

UNIVERSITY OF SURREY

- The data on each edge device may be
  - Class labels are imbalanced
  - Attributes are not independent and identically distributed (Non-IID)
- No all end-users may participate in learning in each round
- The number of edge devices may be huge
- The computation power, storage capacity, energy consumption and communication bandwidth are limited

Layer-wise Asynchronous Federated Learning

Y. Chen, X. Sun, and Y. Jin. Communication-efficient federated deep learning with layer-wise asynchronous model update and temporally weighted aggregation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(10): 4229 – 4238, 2020,

Canonical Federated Learning

UNIVERSITY OF SURREY

- Server Component
  1. Random initialization
  2. T communication rounds
  3. Participating set selection
  4. Call local training
  5. Aggregation

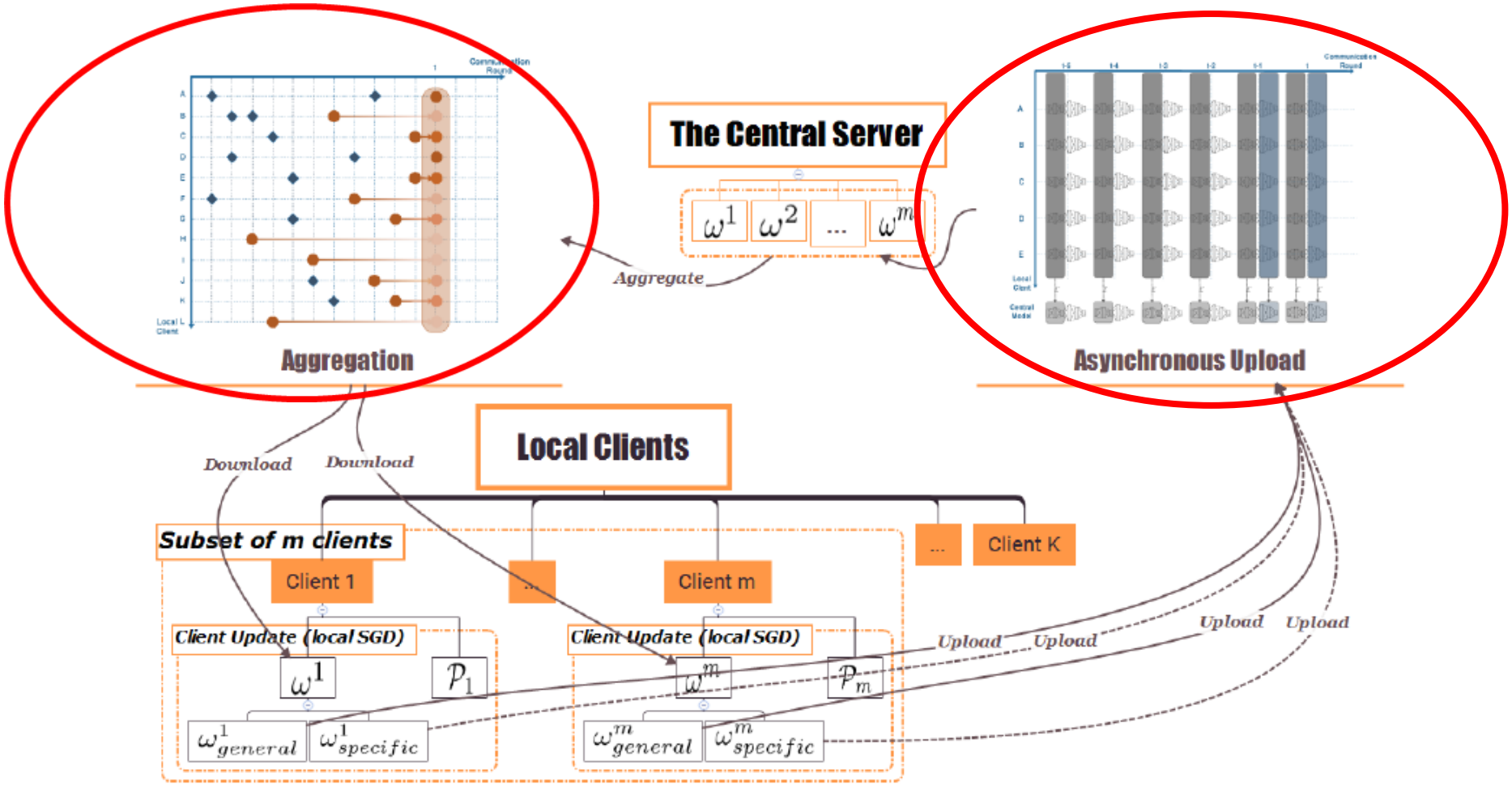
- Client Component
  1. Parameter download
  2. Local stochastic gradient descent
  3. Parameter upload

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots$ **do**
  $m \leftarrow \max(C \cdot K, 1)$
  $S_t \leftarrow$ (random set of m clients)
  for each client $k \in S_t$ **in parallel do**
    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w): *// Run on client k*
$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    $w \leftarrow w - \eta \nabla \ell(w; b)$
return w to server

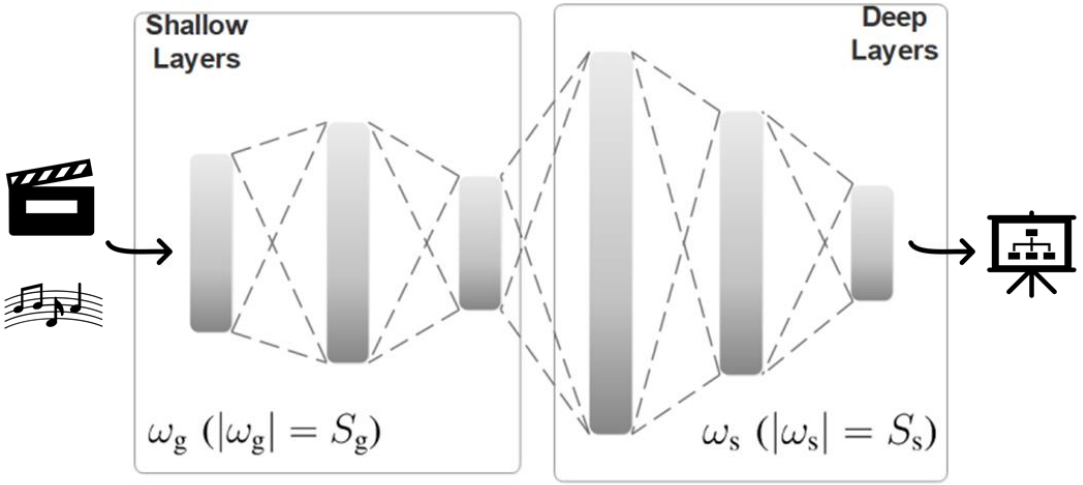
Layer-wise Asynchronous Model Update



1. Layer-wise asynchronous model update
2. Temporally weighted aggregation

Learning Dynamics in DNNs

- Assuming deep neural networks such as convolutional neural networks are used on each device
- Interesting observations:
  - **shallow layers** learn the general features applicable to distinct tasks and datasets
  - **deep layers** learn specific features that are significantly related to specific data, e.g., data stored in local clients
- Parameters of the shallow layers should be updated more frequently than those of in the deep layers



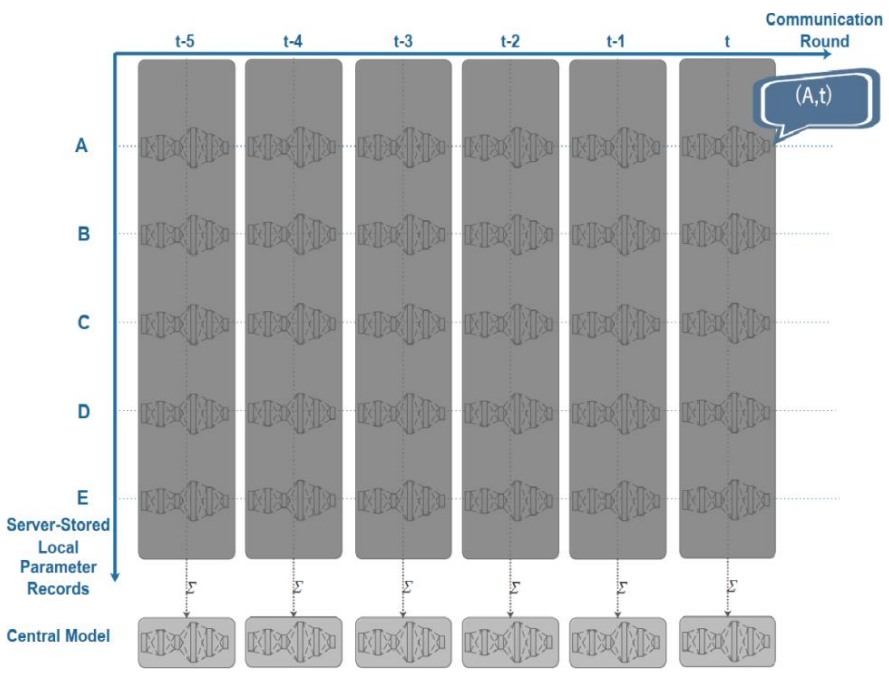
$$S_g \ll S_s$$

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *NIPS*'2014

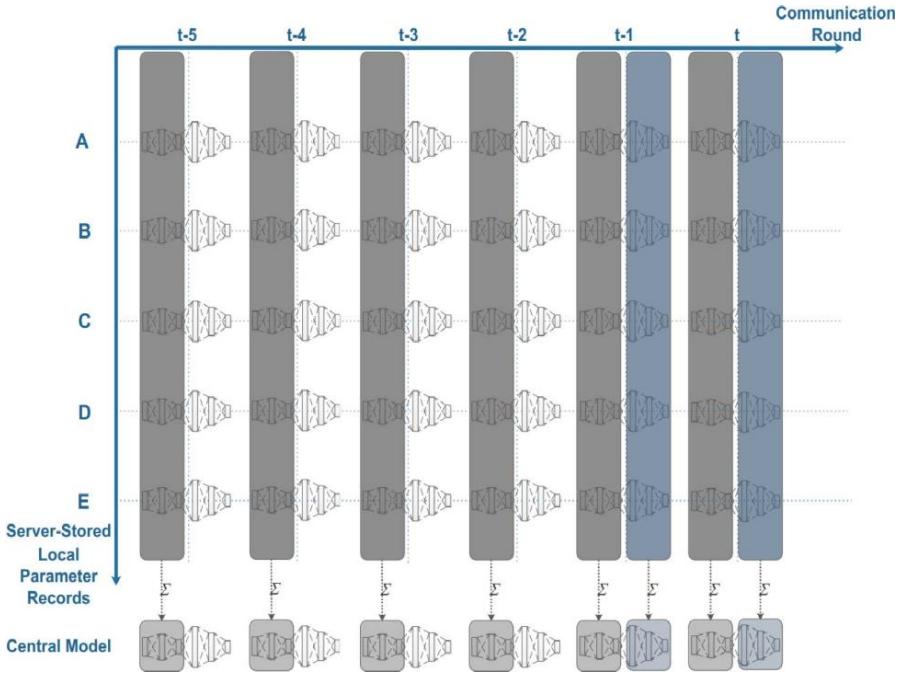
Layer-wise Asynchronous Updating

- Since the weigh updates from different devices are updated in different time instants, a temporally weighted aggregation strategy is also introduced

FedAVG



AS_FedAVG

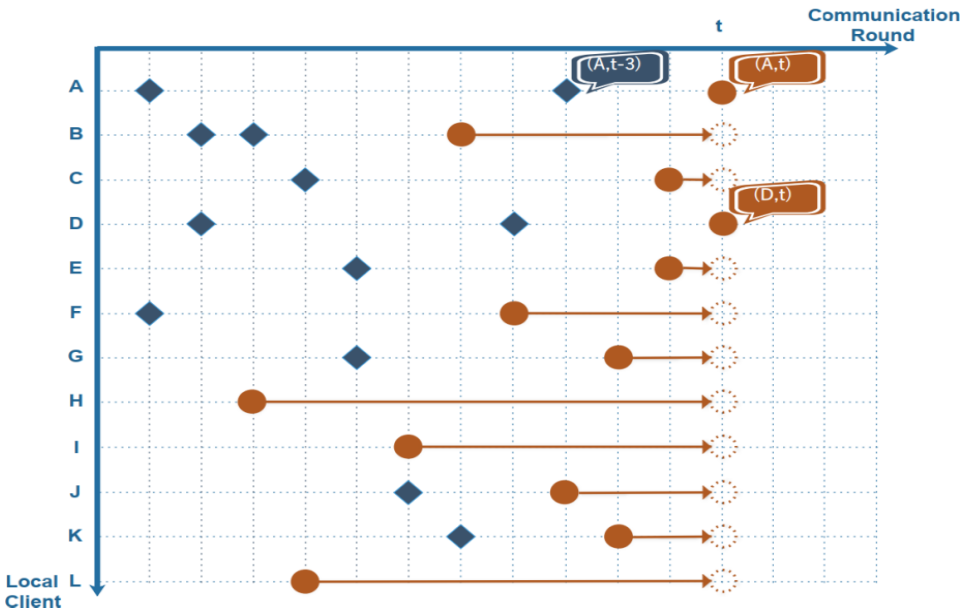


Temporally Weighted Aggregation

- Since the weigh updates from different devices are updated in different time instants, a temporally weighted aggregation strategy is also introduced

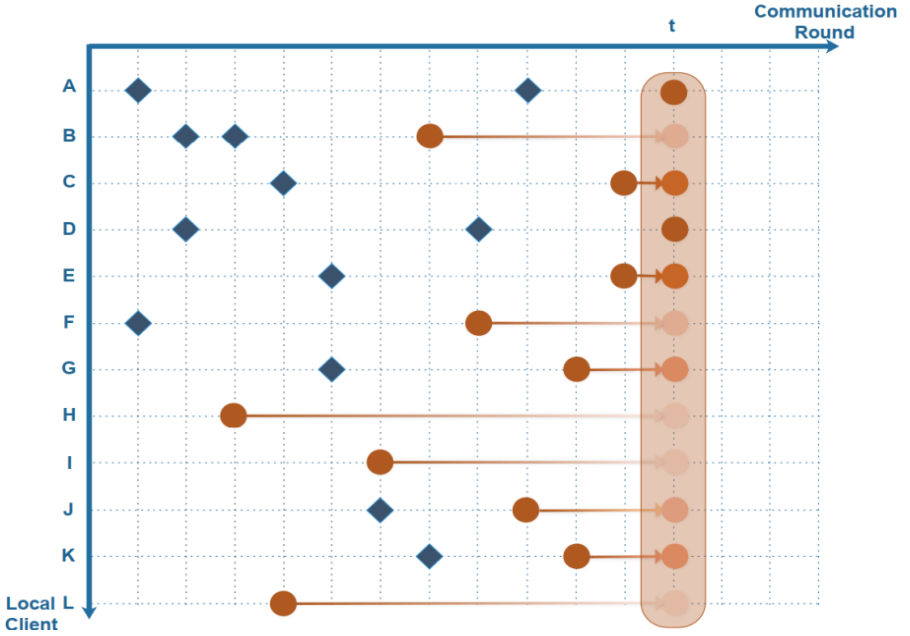
FedAVG

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$$



TW_FedAVG

$$\omega_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} (e/2)^{-(t-\text{timestamp}^k)} \omega^k$$



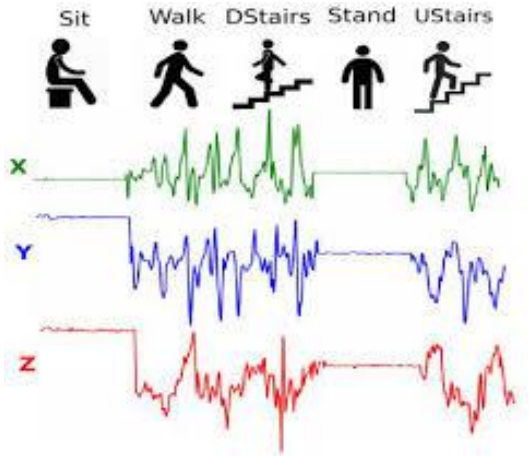
Empirical Results

- **CNN** with two stacked convolution layers and one fully connected layer for the **MNIST** handwritten digit recognition task

PARAMETERS SETTINGS FOR THE CNN

Layer	Shapes
conv2d_1	$5 \times 5 \times 1 \times 32$
conv2d_1	32
conv2d_2	$5 \times 5 \times 32 \times 64$
conv2d_2	64
dense_1	1024×512
dense_1	512
dense_2	512×10
dense_2	10

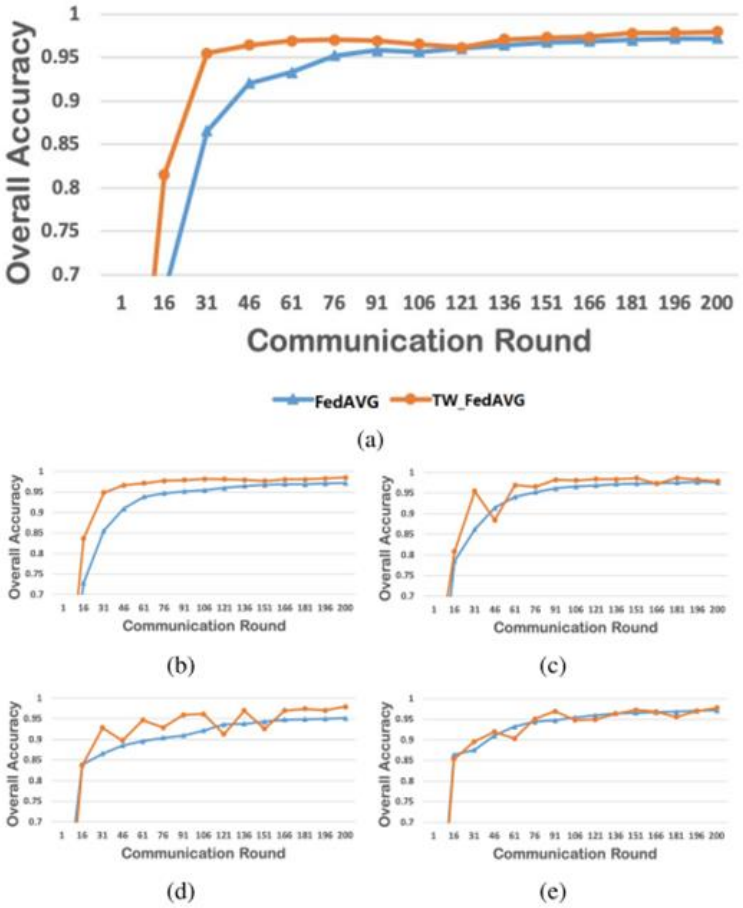
- **LSTM** with two stacked LSTM layers and one fully connected layer for the **human activity recognition (HAR)** task



PARAMETER SETTINGS FOR THE LSTM

Layer	Shapes
lstm_1	9×100
lstm_1	25×100
lstm_1	100
lstm_2	25×100
lstm_2	25×100
lstm_2	100
dense_1	25×256
dense_1	256
dense_2	256×6
dense_2	6

Empirical Results – CNN on MNIST Data



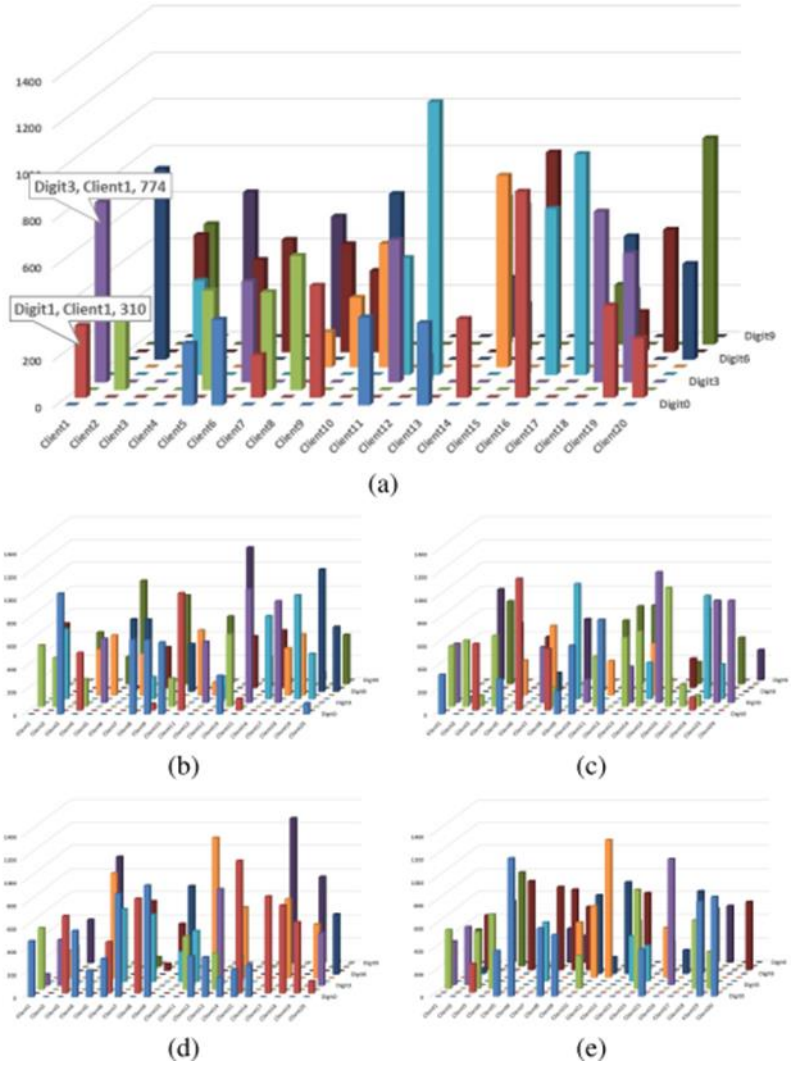


Fig. 7. Comparative studies on temporally weighted aggregation on data set MNIST using the CNN. (a) 1@MNIST. (b) 2@MNIST. (c) 3@MNIST. (d) 4@MNIST. (e) 5@MNIST.

Accuracy & Communication Costs

UNIVERSITY OF SURREY

EXPERIMENTS ON PERFORMANCE

Dataset_ID@Task	FedAVG		TW_FedAVG			ASTW_FedAVG		AS_FedAVG	
	*Round (Accuracy)**	*C. Cost***	*Round (Accuracy)**		*C. Cost***	*Round (Accuracy)**	*C. Cost***	*Round (Accuracy)**	*C. Cost***
1@MNIST*	75 (97.2%)	6.16	**31 (97.9%)**		0.74	106 (97.7%)	1	175 (95.4%)	2.46
2@MNIST*	85 (97.2%)	6.76	**32 (98.5%)**		1.33	61 (98.1%)	**1**	—(94.8%)†	—†
3@MNIST*	73 (97.7%)	5.99	**31 (98.7%)**		1.13	70 (97.9%)	**1**	—(94.9%)†	—†
4@MNIST*	196 (95.2%)	8.18	**61 (98.0%)**		1.14	136 (96.1%)	**1**	—(93.1%)†	—†
5@MNIST*	98 (97.1%)	3.28	**76 (97.8%)**		3.17	61 (96.1%)	**1**	109 (96.7%)	2.66
1@HAR**	526 (92.3%)	4.72	**166** (94.0%)		**0.69**	358 **(94.6%)**	1	—(89.2%) ‡	—‡
2@HAR**	451 (94.7%)	5.65	**119** (95.4%)		**0.57**	313 **(95.9%)**	1	—(82.9%) ‡	—‡
3@HAR**	—(87.3%) ‡	—‡	**174 (94.4%)**		**0.69**	376 (93.2%)	1	—(88.5%) ‡	—‡
4@HAR**	856 (90.2%)	7.05	**181 (95.1%)**		1.28	211 (94.1%)	**1**	751 (91.2%)	5.30
5@HAR**	571 (92.6%)	5.49	**155 (93.6%)**		**0.57**	404 (93.1%)	1	646 (92.5%)	2.38

* Cells are filled when the accuracy reaches 95% within 200 rounds.
** Cells are filled when the accuracy reaches 90% within 1000 rounds.
† Cells are marked when it can not reach 95% within 200 rounds.
‡ Cells are marked when it can not reach 90% within 1000 rounds.
* Rounds are needed to reach a certain accuracy, after which the best accuracy (within 200/1000 rounds) is listed in parenthesis.
** Total communication cost; the C. cost of ASTW_FedAVG is termed as 1.

TW_FLAVG: adopts temporally weighted aggregation without the layer-wise asynchronous model update
AS_FedAVG: employs the layer-wise asynchronous model update without using temporally weighted aggregation

Ternary Compression for Communication-efficient Federated Learning

J. Xu, W. Du, Y. Jin, W. He, and R. Cheng. Ternary compression for communication-efficient federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2020 (accepted)

Ternary Weighted Neural Networks

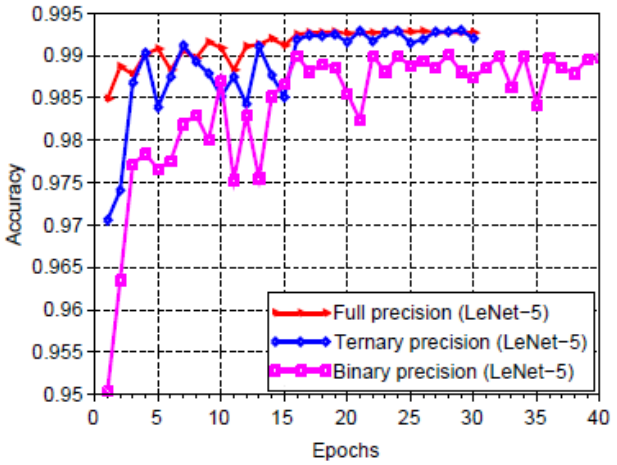
$$\alpha^*, \theta^{t*} = \arg\min_{\alpha,\theta^t} \|\theta - \alpha\theta^t\|_2^2,$$

$$\theta_l^t = \begin{cases} +1, & \theta_l > \Delta_l \\ 0, & |\theta_l| \le \Delta_l \\ -1, & \theta_l < -\Delta_l \end{cases}$$

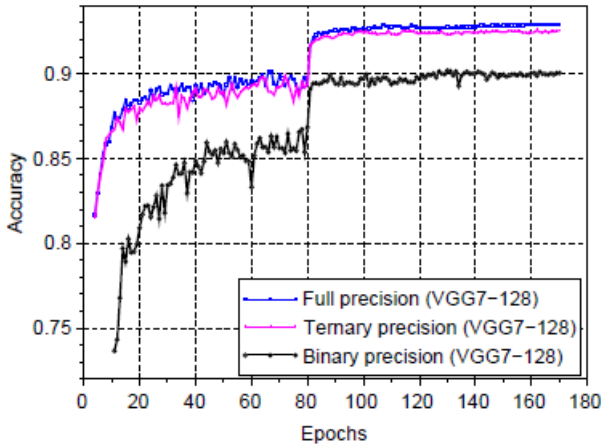
$$\Delta_l^* = \frac{0.7}{d^2} \sum_i^{d^2} (|\theta_l^i|)$$

$$\alpha_l^* = \frac{1}{|I_{\Delta_l}|} \sum_i^{|I_{\Delta_l}|} |\theta_i|$$

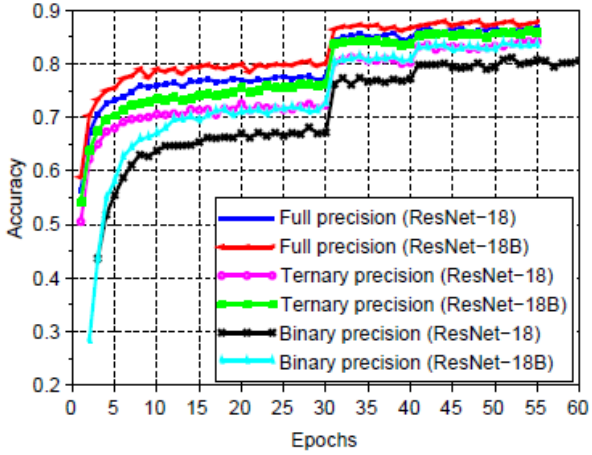
$$I_{\Delta_l} = \{i \mid \theta_i > \Delta_l\}$$



(a) MNIST



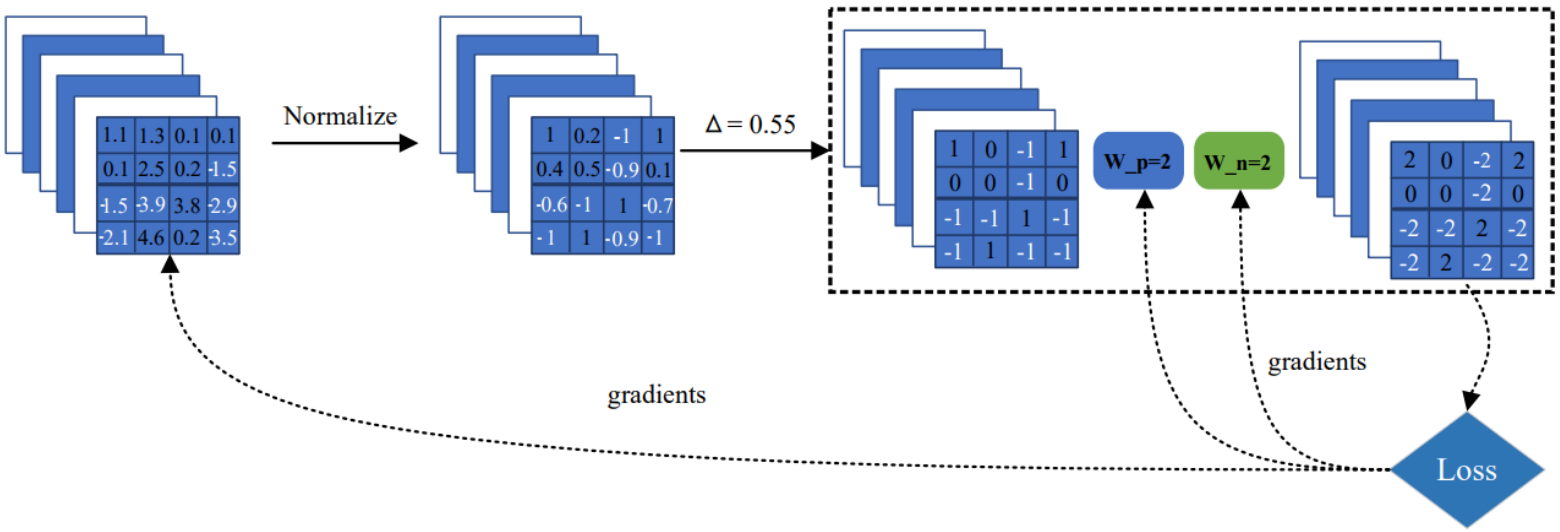
(b) CIFAR-10



(c) ImageNet (top-5)

Li *et al*, Ternary weighted networks. arXiv:1605.04711v2 [cs.CV] 19 Nov 2016.

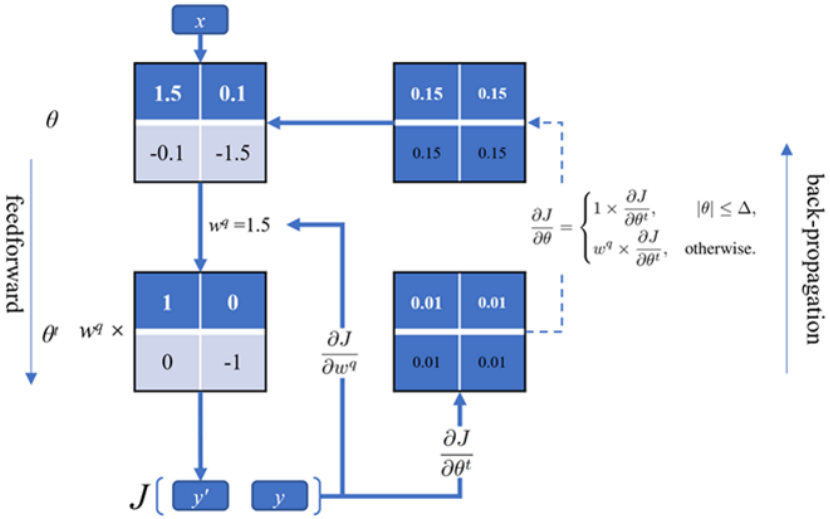
Trained Ternary Quantization



$$\theta_l^t = \begin{cases} w_l^p, \tilde{\theta}_l > \Delta_l \\ 0, \left|\tilde{\theta}_l\right| \leq \Delta_l \\ -w_l^n, \tilde{\theta}_l < -\Delta_l \end{cases}$$

$$\Delta_l = t \times \max(|\theta_l|)$$

t is a hyper-parameter



Zhu *et al*. Trained ternary quantization. arXiv preprint arXiv:1612.01064, 2017 (ICLR 2017)

Federated Trained Ternary Quantization

UNIVERSITY OF SURREY

- Normalize layer weights to [-1, 1]: $\theta^s = g(\theta)$
- Calculate threshold for client k: $\Delta = T_k \times \max(|\theta^s|)$

where $T_k = \begin{cases} 0.05 + 0.01 \times \text{rand}(0,1), & \text{if} \quad \text{rand}(0,1) > 0.5 \\ 0.05 + 0.01 \times \frac{k}{N}, & \text{if} \quad \text{rand}(0,1) \leq 0.5 \end{cases}$

- Calculate threshold using sparsity:

$$\Delta = \frac{T_k}{d^2} \sum_{i}^{d^2} (|\theta_i^s|)$$

where the relationship below holds:

$$\Delta = T_k \times \frac{1}{d^2} \sum_{i}^{\tilde{}} (|\theta_i^s|) \leq T_k \times \frac{1}{d^2} (d^2 \times \max|\theta^s|)$$

$$\leq T_k \times \frac{1}{d^2} (d^2 \times 1) \leq T_k,$$

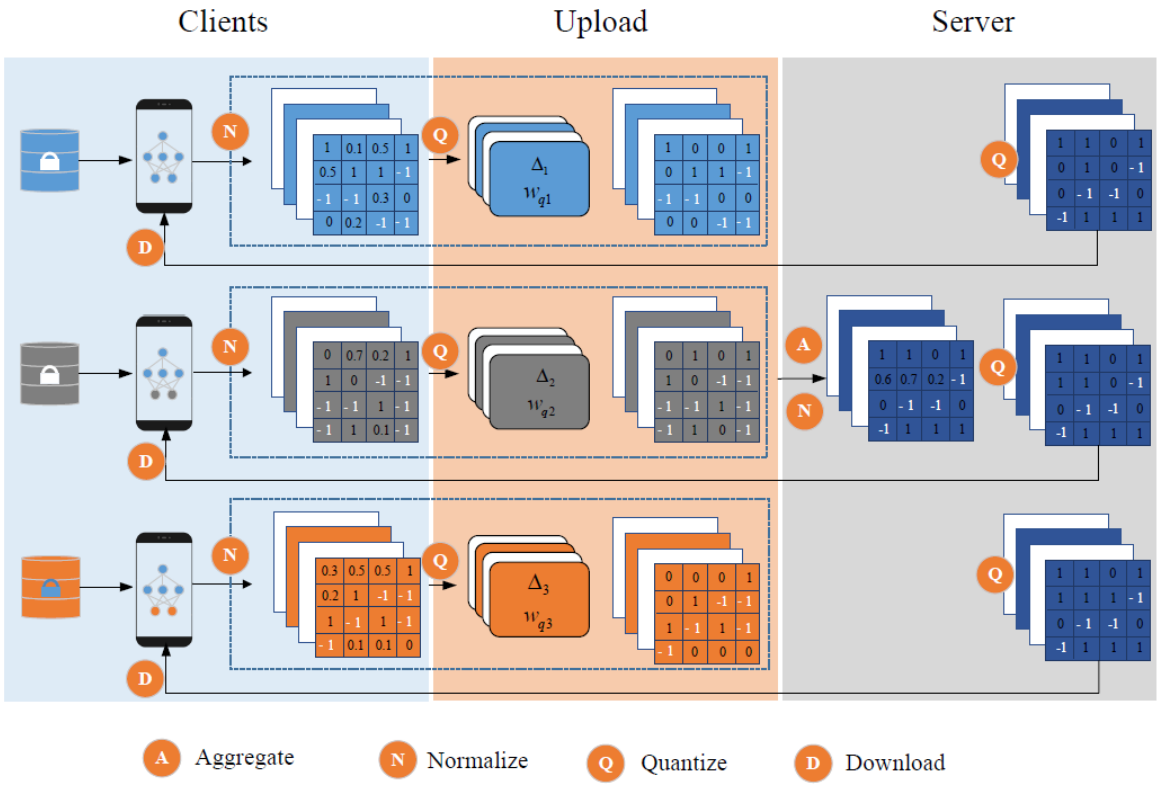
If T_k is 0.7, it is the same with TWN.

Algorithm 1: Federated Trained Ternary Quantization (FTTQ)

Input: Full-precision parameters θ and quantization vector w^q, loss function l, dataset D with sample pairs $(x_i, y_i), i = \{1, 2, ..., |D|\}$, learning rate η.
Output: Quantified model θ^t
init: All clients parameters are initialized with θ.
for $(x_i, y_i) \in D$ **do**
- $\theta^s \leftarrow g(\theta)$
- $mask \leftarrow \varepsilon(|\theta^s| - \Delta)$
- $I^t = \text{sign}(mask \odot \theta^s)$
- $\theta^t \leftarrow w^q \times I^t$
- $J \leftarrow l_i(x_i, y_i; \theta^t)$
- $\frac{\partial J}{\partial w^q} \leftarrow \sum_{i \in I_p} \frac{\partial J}{\partial \theta_i^t}$
- $w^q \leftarrow w^q + \eta \frac{\partial J}{\partial w^q}$
- $\theta^t \leftarrow \theta^t + \eta \frac{\partial J}{\partial \theta^t}$
- update θ

end
Return θ^t (including w^q, I^t)

Ternary Federated Averaging (T-FedAvg)



Algorithm 2: Ternary Federated Averaging

Input: Initial global model parameters θ_0
Init: Broadcast the global model θ_0 to all clients
for round r = 1,..., T **do**
  for Client $k \in \mathbb{K} = \{1, 2, ..., |\lambda N|\}$ **in parallel do**
    load dataset D_k
    $\theta_k \leftarrow \theta_{r-1}^t$ or θ_{r-1}
    initialize w^q
    $\theta_{k,r}^t \leftarrow$ **FTTQ**(θ_k, w^q)
    upload $\theta_{k,r}^t$ to server
  end
  Server does:
    $\theta_r \leftarrow \sum_{k=1}^{\lambda N} \frac{|D_k|}{\sum_{k=1}^{\lambda N} |D_k|} \theta_{c,r}^t$
    $\Delta^S = 0.05 \times \max(|\theta_r|)$
    $I_p^S,\ I_n^S = \{i \mid \theta_r^i > \Delta^S\},\ \{i \mid \theta_r^i < -\Delta^S\}$
    $w_p^S,\ w_n^S = \frac{1}{|I_p^S|} \sum_i^{|I_p^S|} (|\theta_r^i|),\ \frac{1}{|I_n^S|} \sum_i^{|I_n^S|} (|\theta_r^i|)$
    $\theta_r^t \leftarrow w_p^S \times I_p^S - w_n^S \times I_n^S$
    if θ_r^t *does not crash* **then**
      broadcast θ_r^t /*Strategy I*/
    else
      broadcast θ_r /*Strategy II*/
    end
  end
end

Experimental Settings

1) **Compared algorithms**. In this work, we compare the following algorithms:

- Baseline: the centralized learning algorithm, such as stochastic gradient descent (SGD) method, which means that all data is stored in a single computing center and the model is trained directly using the entire data.
- FedAvg: the canonical federated learning approach presented in [16].
- CMFL: a communication-mitigated federated learning algorithm proposed in [36].
- T-FedAvg: our proposed quantized federated learning approach. Note that the first and last feature layers of the global and local models are full-precision.

TABLE I
MODELS AND HYPERPARAMETERS.

Models	MLP	CNN	ResNet*
Dataset	MNIST	CIFAR10	CIFAR10
Optimizer	SGD	Adam	Adam
Learning rate	0.01	0.001	0.008
Basic accuracy(%)	92.25 $\pm$ 0.06	85.63 $\pm$ 0.10	88.80 $\pm$ 0.20

- Total number of clients: $N = 100$ for MNIST with simulation, and $N = 5$ for CIFAR10 (as suggested in [51]) using the physical system.
- The participation ratio per round: $\lambda = 0.1$ for MNIST and $\lambda = 1$ for CIFAR10.
- Classes per client: $N_c = 10$.
- Local epochs: $E = 5$ for MNIST, and $E = 10$ for CIFAR10.

Experimental Results – IID Data

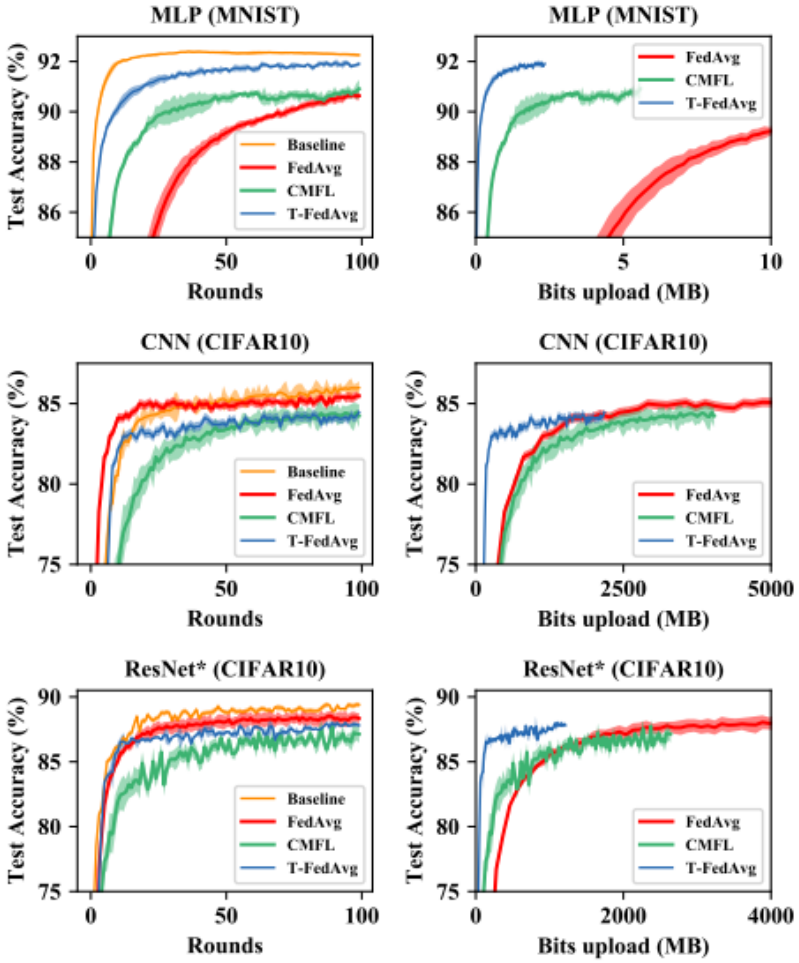


Fig. 4. Convergence curves over rounds and bits upload of the compared algorithms. For each algorithm, the solid line denotes the averaged test accuracy over five independent runs, and the shaded region denotes the standard deviation.

TABLE II

THE COMMUNICATION COSTS IN 100 ROUNDS ON IID DATA. FOR MLP, 10 OUT OF 100 CLIENTS ($\lambda = 0.1$) PARTICIPATE IN EACH ROUND OF TRAINING. FOR CIFAR10, ALL FIVE CLIENTS PARTICIPATE IN EACH ROUND OF TRAINING ($\lambda = 1.0$).

Methods	MLP	CNN	ResNet*
	upload / download (MB)	upload / download (MB)	upload / download (MB)
FedAvg	19.53 / 19.53	16196.59 / 16196.59	9253.81/ 9253.81
CMFL	5.63 / 19.53	4079.75 / 16196.59	2666.80 / 9253.81
T-FedAvg	2.36 / 2.36	2201.84 / 3041.53	1229.27 / 6847.82

TABLE III

TEST ACCURACIES ACHIEVED AND WEIGHTS WIDTH OF DIFFERENT ALGORITHMS WHEN TRAINED ON IID DATA

Accuracy	MNIST	CIFAR10		Width
	MLP	CNN	ResNet*	bit
Baseline	92.25 ± 0.06	85.63 ± 0.10	88.80 ± 0.20	32
FedAvg	90.63 ± 0.17	85.47 ± 0.14	88.34 ± 0.40	32
CMFL	90.91 ± 0.36	84.23 ± 0.71	87.13 ± 0.66	32
T-FedAvg	91.95 ± 0.11	84.46 ± 0.17	87.87 ± 0.18	2

Experimental Results – Non-IID Data

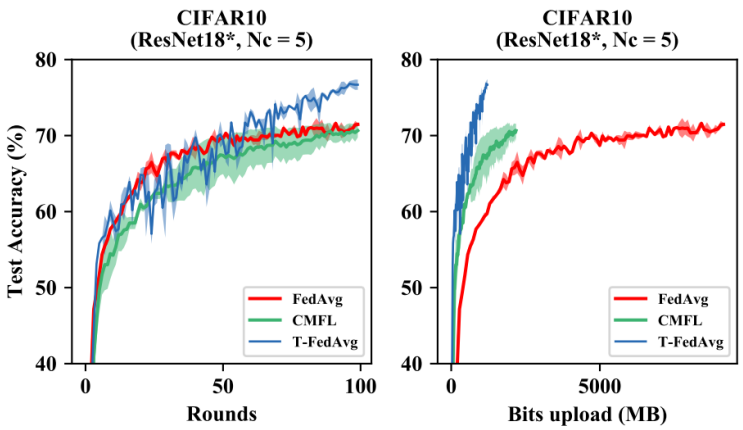


Fig. 6. Convergence trends and communication costs of FedAvg, CMFL and T-FedAvg when $N_c = 5$.

TABLE IV
TEST ACCURACIES ACHIEVED OVER NON-IID DATA FOR DIFFERENT N_c.

Accuracy (%)	MNIST		CIFAR10 (ResNet*)	
	$N_c = 2$	$N_c = 5$	$N_c = 2$	$N_c = 5$
FedAvg	82.61 ± 4.31	89.24 ± 0.38	40.17 ± 5.6	71.47 ± 0.23
CMFL	81.51 ± 2.29	88.30 ± 0.52	39.30 ± 6.1	70.26 ± 0.63
T-FedAvg	87.29 ± 0.89	90.04 ± 0.68	40.46 ± 4.3	76.69 ± 0.67

Evolutionary Multi-Objective Federated Neural Structure Optimization

H. Zhu and **Y. Jin**. Multi-objective evolutionary federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4): 1310-1322, 2020

Bi-Objective Federated Learning

UNIVERSITY OF SURREY

- Objectives
  - Maximization of the learning performance of the central model
  - Minimization of the communication cost
- Decision variables
  - The hyperparameters, such as learning rate, batch size
  - Parameters of the deep neural network
  - Structure of the deep neural network
- How to encode deep neural networks such as CNN and MLP?

Scalable Encoding of Neural Connectivity

UNIVERSITY OF SURREY

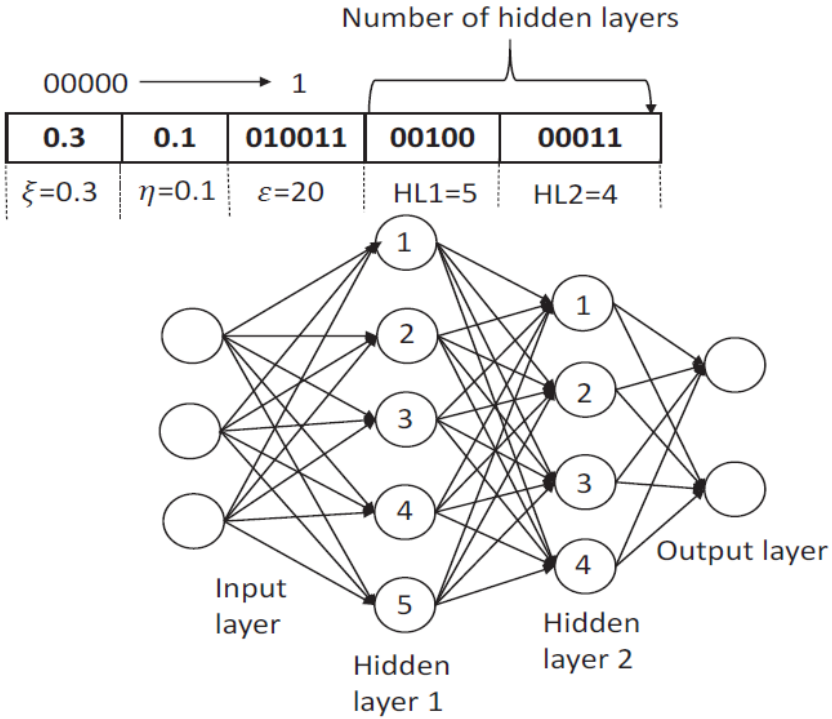
- Encoding of deep neural networks is extremely challenging since it involves a very large number if decision variables
- A modified sparse evolutionary training (SET) is adopted:
  - Use the Erdös Renyi random graph random graph to determine the connectivity between every two neighboring layers of the neural network

$$p(W_{ij}^k) = \frac{\varepsilon(n^k + n^{k-1})}{n^k n^{k-1}}$$
$$n^W = n^k n^{k-1} p(W_{ij}^k)$$

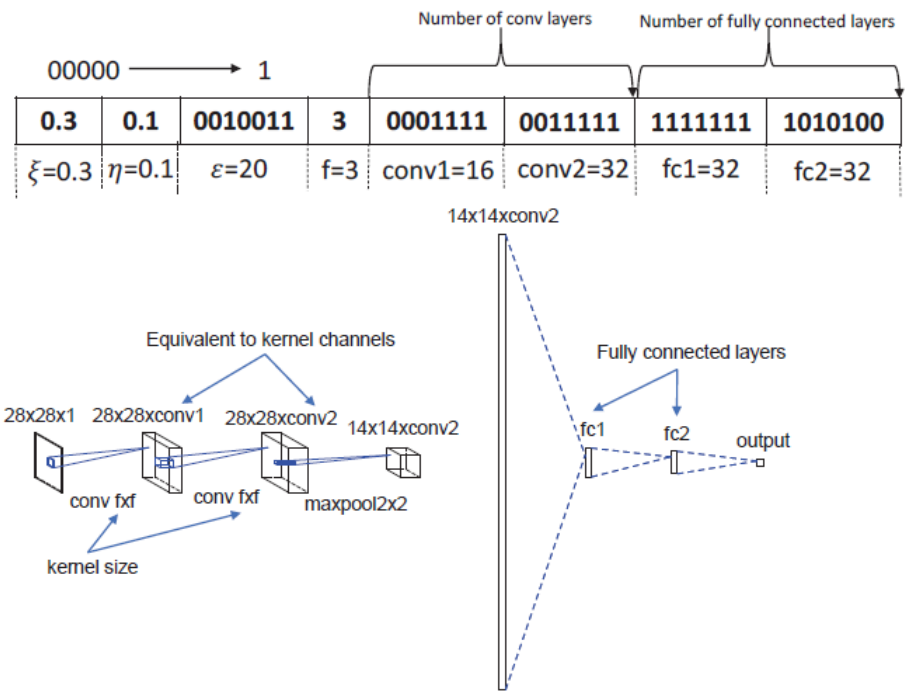
  - where n^k and n^{k-1} are the number of neurons in layer k and $k-1$, respectively, W^k_{ij} is the sparse weight matrix between the two layers, ε is a SET parameter that controls connection sparsity, and n^W is the total number of connections between the two layers
  - It is easy to find that the connection probability would become significantly lower, if $\varepsilon \ll n^k$ and $\varepsilon \ll n^{k-1}$
  - Remove a fraction ξ of the weights that have updated the smallest during each training epoch, which can be seen as the selection operation of an evolutionary algorithm
  - Removal is applied at the last SGD iteration only

D. C. Mocanu, E. Mocanu, P. Stone and A. Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9, Article number: 2383 (2018)

Genetic Representation



MLP



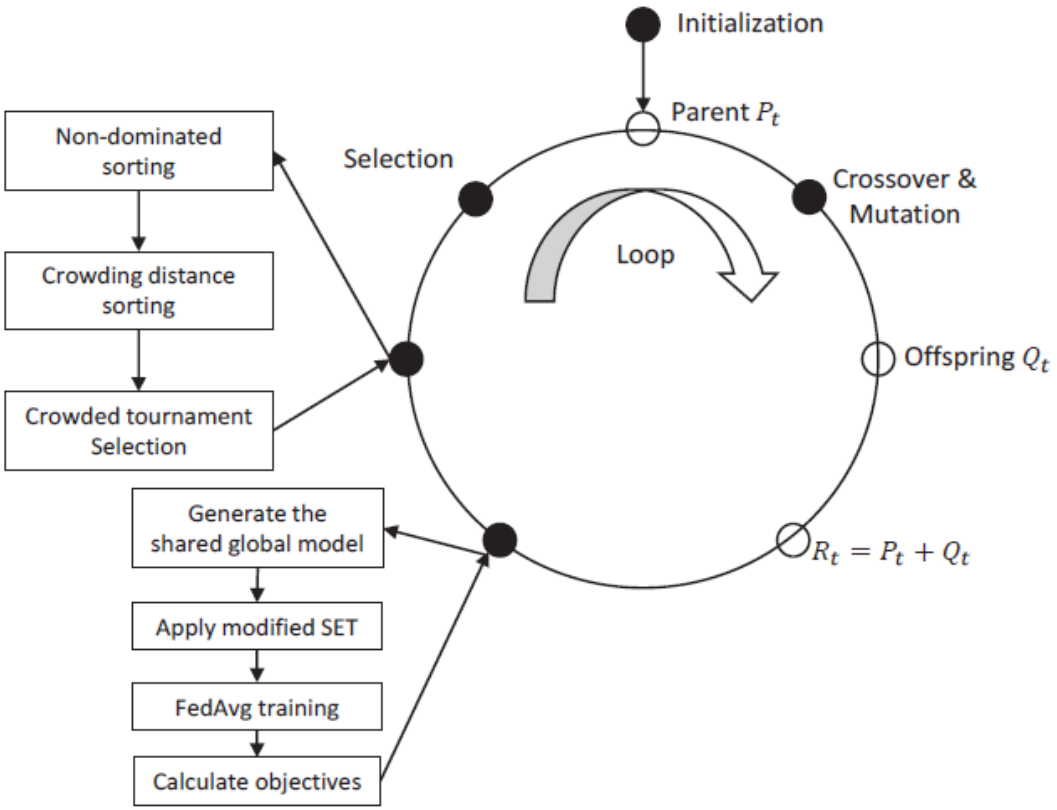
CNN

Bi-Objective Federated Learning

- Minimize the following two objectives using NSGA-II

$$E_t = 1 - A_t$$

$$\Omega_t = \sum_{k=1}^{K} \Omega_k / K$$

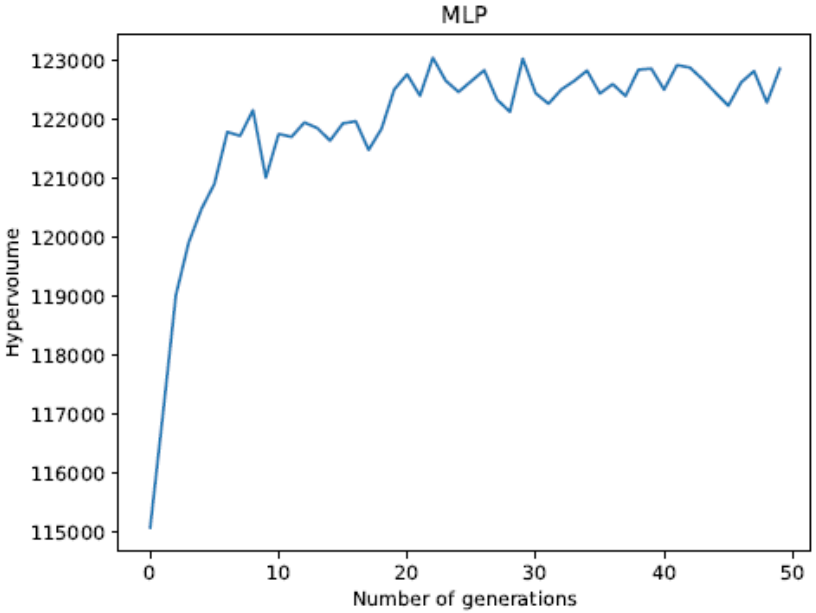


Experimental Settings

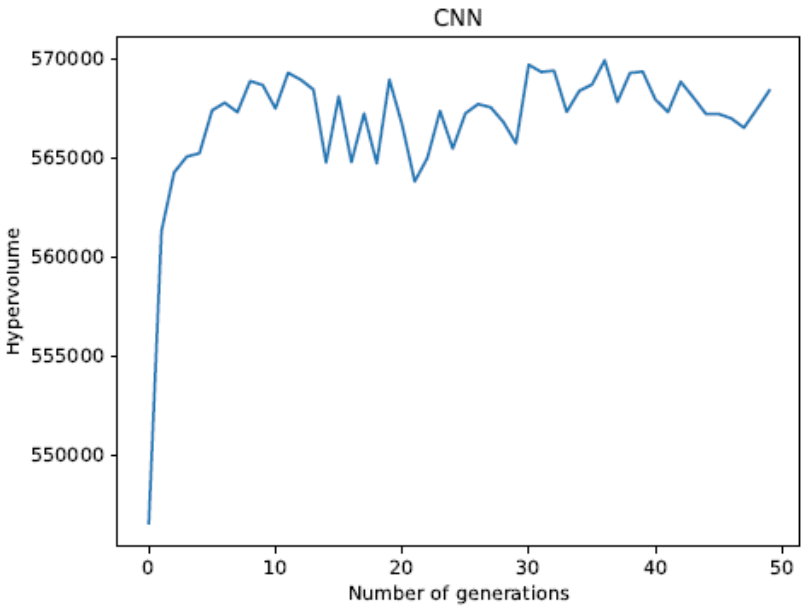
UNIVERSITY OF SURREY

- The **standard FL**: MLP and CNN on the MNIST data
  - MLP: a learning rate of 0.1 and the batch size is 50;
  - two hidden layers, each having 200 nodes (**199,210** parameters in total) and uses the ReLu function as the activation function
  - CNN: two 3 × 3 kernel filters (the first with 32 channels and the second with 64 channels)
  - a 2 × 2 max-pooling layer, a 128 fully connected layer and finally a 10 class softmax output layer (**1,625,866** parameters in total)
  - 100 clients, mini-batch size = 50, training epoch = 5
- For the evolutionary FL:
  - Population size =20, generation = 20 for IID data and 50 for non-IID data
  - Communication round = 5 for IID data and 10 for non-IID data

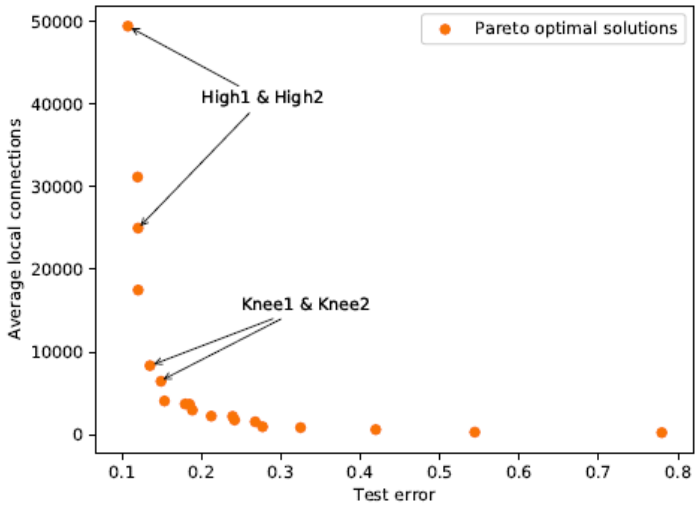
Results

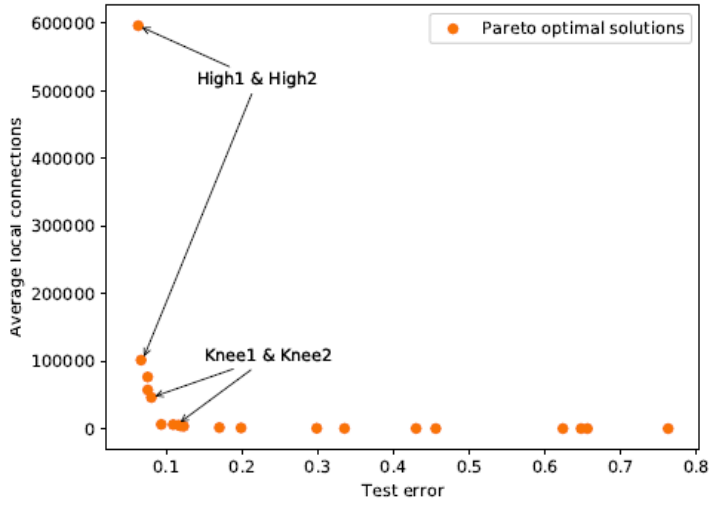


(a) Hypervolume for MLP



(b) Hypervolume for CNN





Results

UNIVERSITY OF SURREY

HYPER-PARAMETERS OF HIGH1, HIGH2, KNEE1, AND KNEE2 FOR ***MLPs*** EVOLVED ON ***non-IID*** DATA AND THEIR VALIDATION RESULTS

Parameters	Knee1	Knee2	High1	High2	Standard
Hidden layer1	49	53	86	109	200
Hidden layer2	/	/	/	/	200
ε	10	8	66	34	/
ξ	0.1106	0.0764	0.1106	0.1566	/
Learning rate η	0.3	0.2961	0.3	0.3	0.1
Test accuracy IID	96.78%	96.41%	97.82%	97.68%	98.13%
Connections IID	7,749	5,621	45,329	22,210	199,210
Test accuracy nonIID	94.85%	94.88%	97.32%	96.21%	97.04%
Connections nonIID	8,086	6,143	45,530	24,055	199,210

HYPER-PARAMETERS OF HIGH1, HIGH2, KNEE1, AND KNEE2 FOR ***CNNs*** EVOLVED ON ***non-IID*** DATA AND THEIR VALIDATION RESULTS

Parameters	Knee1	Knee2	High1	High2	Standard
Conv layer1	17	5	53	33	32
Conv layer2	/	/	/	/	64
Fully connected layer1	29	21	208	31	128
Fully connected layer2	/	/	/	/	/
Kernel size	5	5	5	5	3
ε	18	8	66	20	/
ξ	0.1451	0.1892	0.0786	0.1354	/
Learning rate η	0.2519	0.2388	0.2776	0.2503	0.1
Test accuracy IID	98.84%	98.15%	99.06%	98.93%	98.85%
Connections IID	48949	6262	622090	107224	1,625,866
Test accuracy nonIID	97.92%	97.7%	98.52%	98.46%	98.75%
Connections nonIID	39457	6804	553402	90081	1,625,866

Realtime Federated Neural Architecture Search

- H. Zhu and Y. Jin. Real-time Federated Evolutionary Neural Architecture Search. https://arxiv.org/abs/2003.02793
- H. Zhu, H. Zhang, Y. Jin. From federated learning to federated neural architecture search: A survey. *Complex & Intelligent Systems*, 2020 (accepted)

Neural Architecture Search

UNIVERSITY OF SURREY

How many layers ?

How many channels?

What operations（convolution、pooling、attention…）?

How to connect with other layers ?

How to prepare training, validation and test data ?

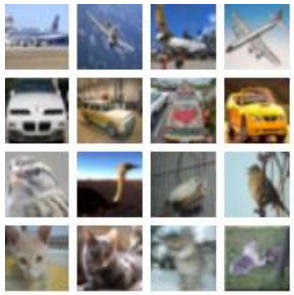
Loss function ?

⋮

Hand-crafted

Trial and errors ?

Neural Architecture Search

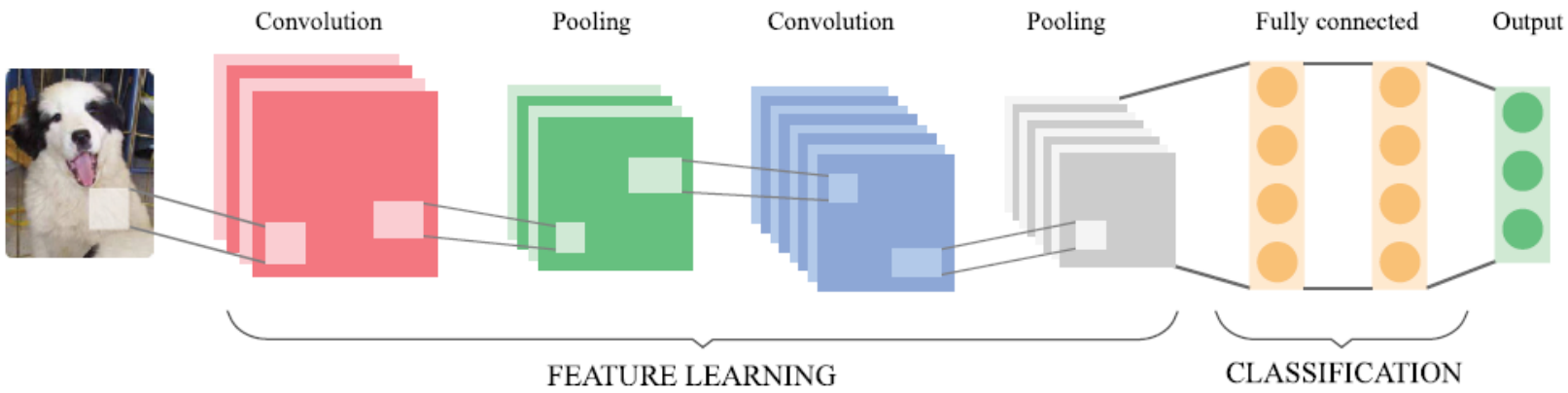


Search Space
(macro/micro)

Search Method
(EAs/RL/GD)

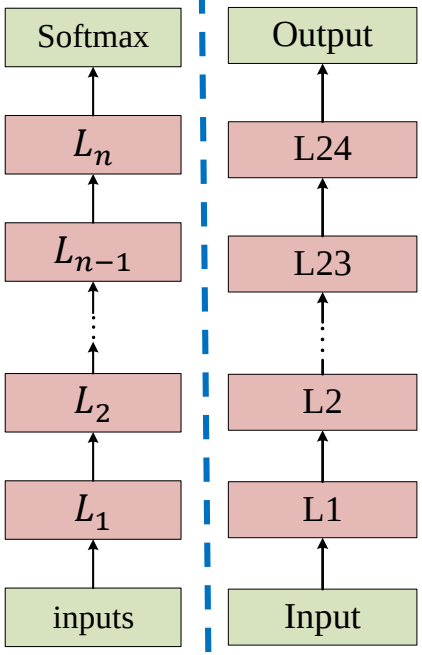
updating

Evaluation
(Single/Multi-Objective)



Neural Architecture Search

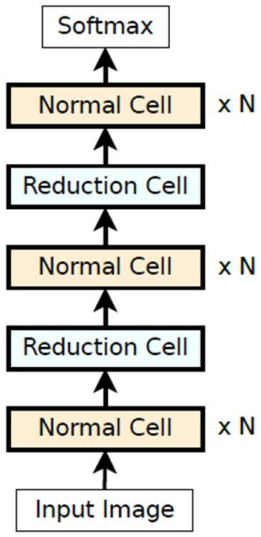
- Huge search space

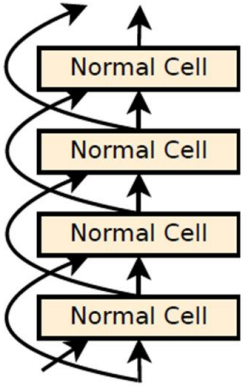


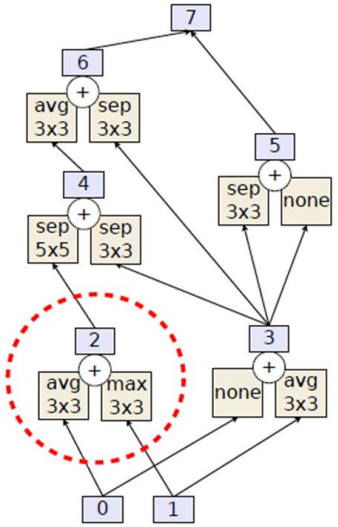
Operation space
3 × 3 conv
5 × 5 conv
7 × 7 conv
3 × 3 max pooling
3 × 3 avg pooling
3 × 3 separable conv
5 × 5 separable conv
7 × 7 separable conv

Total number of neural architectures：

$$24^8 \sim 1 \times 10^{11}$$





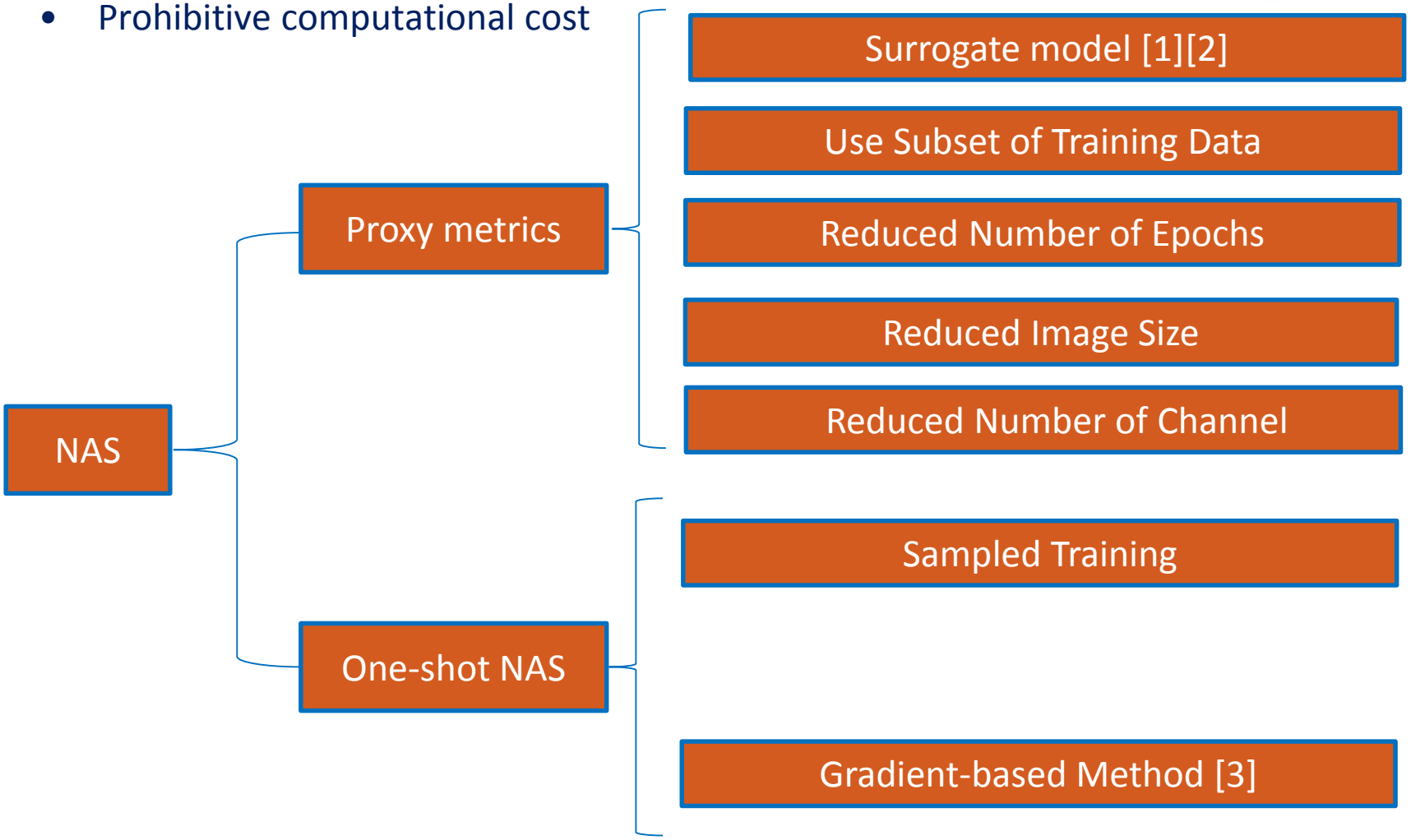


Macro search space

Micro search space

Neural Architecture Search

- Prohibitive computational cost



[1] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen. Data-driven evolutionary optimization: An overview and case studies. *IEEE Transactions on Evolutionary Computation*, 23(3): 442-458, 2019

[2] Y. Sun, H. Wang, B. Xue, Y., G. G. Yen, and M. Zhang. Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. *IEEE Transactions on Evolutionary Computation*, 24(2):350-364, 2020

[3] H. Zhang, Y. Jin, R. Cheng, and K. Hao. Efficient evolutionary search of attention convolutional networks via sampled training and node inheritance. *IEEE Transactions on Evolutionary Computation*, 2020 (accepted)

Federated Neural Architecture Search

UNIVERSITY OF SURREY

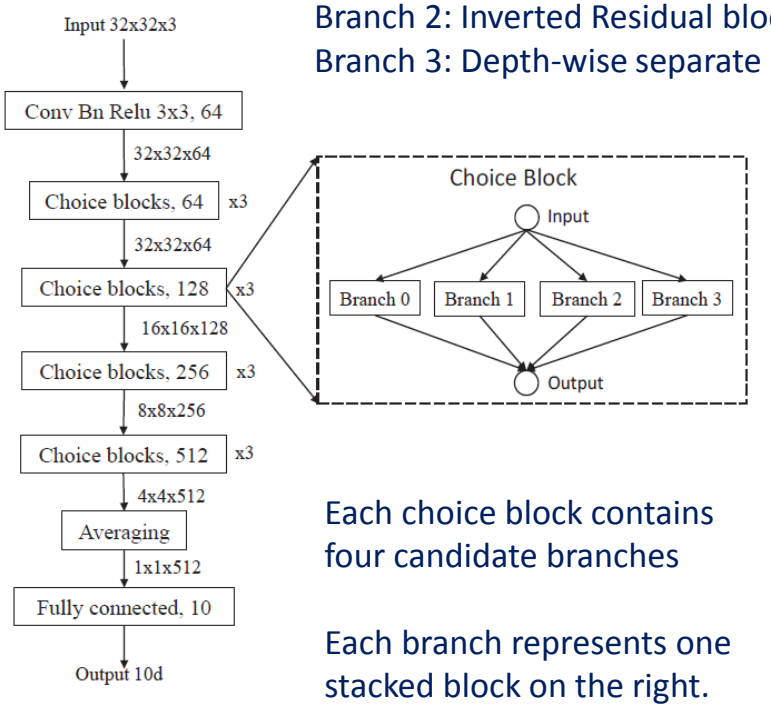
- Main challenges:
  - Computational efficiency
  - Communication cost
  - Online performance requirements
    - Serious performance degradation is not allowed
    - A client is allowed to evaluate one neural architecture in one round

- Solution – **Double sampling**:
  - Model sampling:
    - Sample only one branch path of all the choice blocks in the supernet to randomly generate one sub-network
    - Each client only download the sampled sub-network for local training to reduce both communication and computation time
  - Client sampling: Randomly sample clients for each individual to enable completing the fitness evaluations within one communication round

Federated Neural Architecture Search

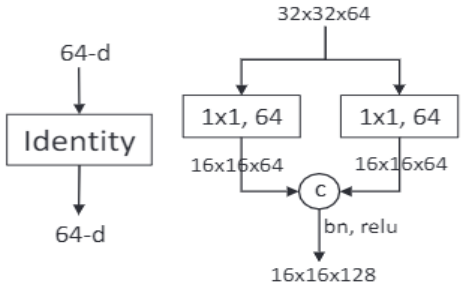
UNIVERSITY OF SURREY

Branch 0: Identity block
Branch 1: Residual block
Branch 2: Inverted Residual block
Branch 3: Depth-wise separate block

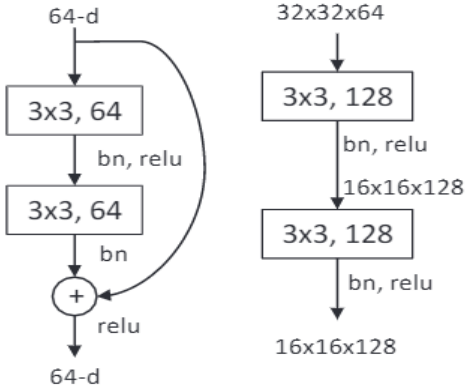


Each choice block contains four candidate branches

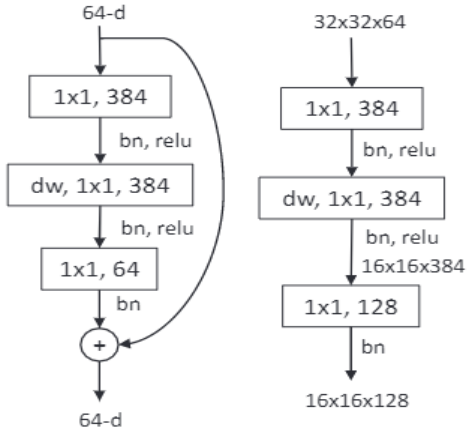
Each branch represents one stacked block on the right.



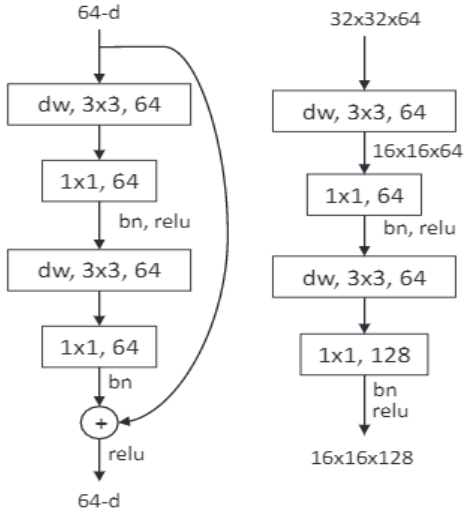
(a) Identity Block



(b) Residual Block



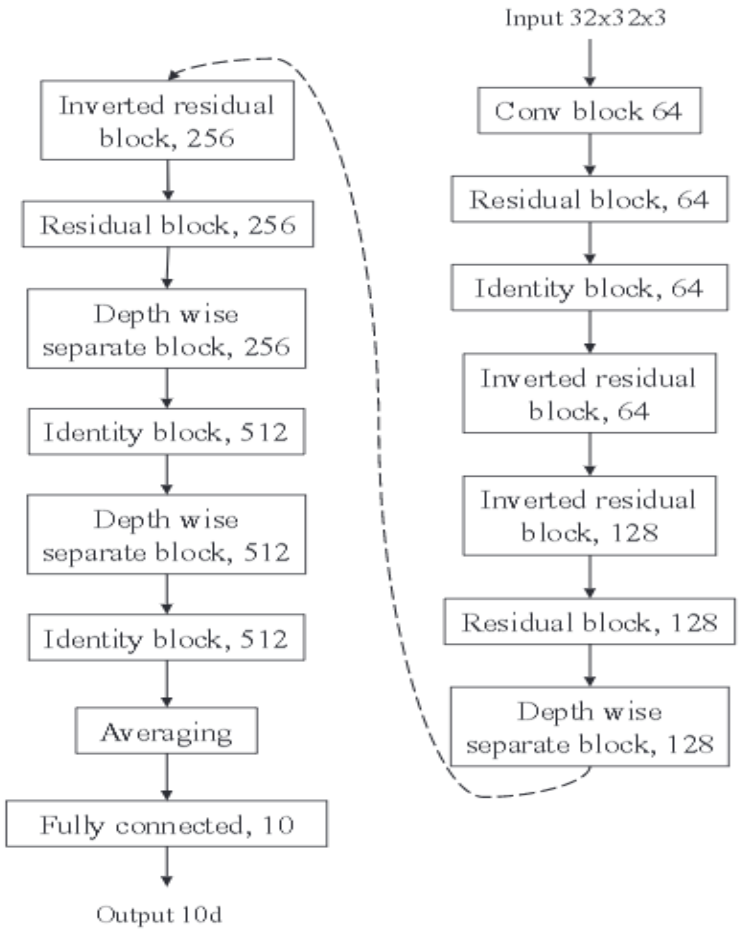
(c) Inverted Residual Block

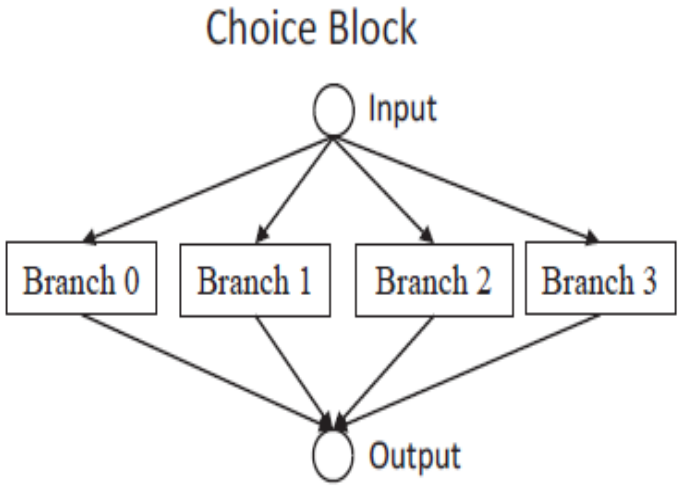


(d) Depth Wise Separate Block

Federated Neural Architecture Search

- Two bit encoding is enough (i.e. 00->Identity block, 01->Residual block)
- A choice key [0 1, 0 0, 1 0, 1 0, 0 1, 1 1, 1 0, 0 1, 1 1, 0 0, 1 1, 0 0] represents a ***sampled*** sub network which is shown below:



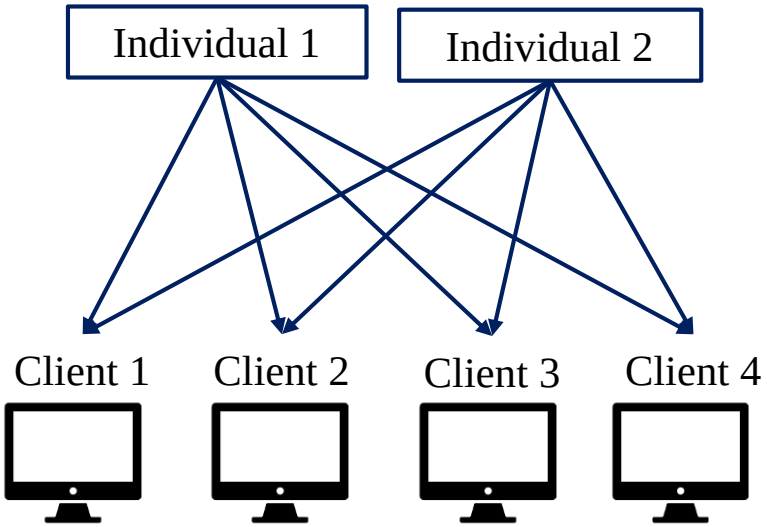


Bit code [0, 0] means branch 0 is sampled in the choice block

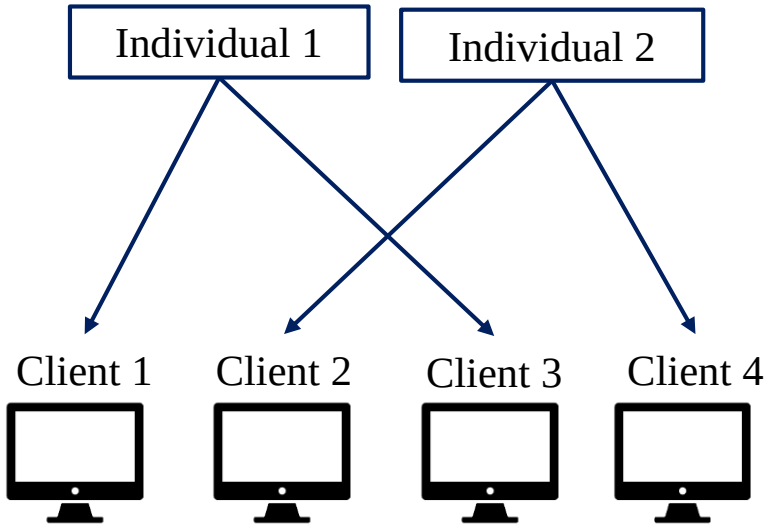
Federated Neural Architecture Search

- Randomly sample a subset of the connected clients for fitness evaluation of one individual
- Clients for the same individual downloads and updates the same sub-model. For instance, if the population size is 2, we have total 4 connected clients, the random partition is shown below:

No client sampling



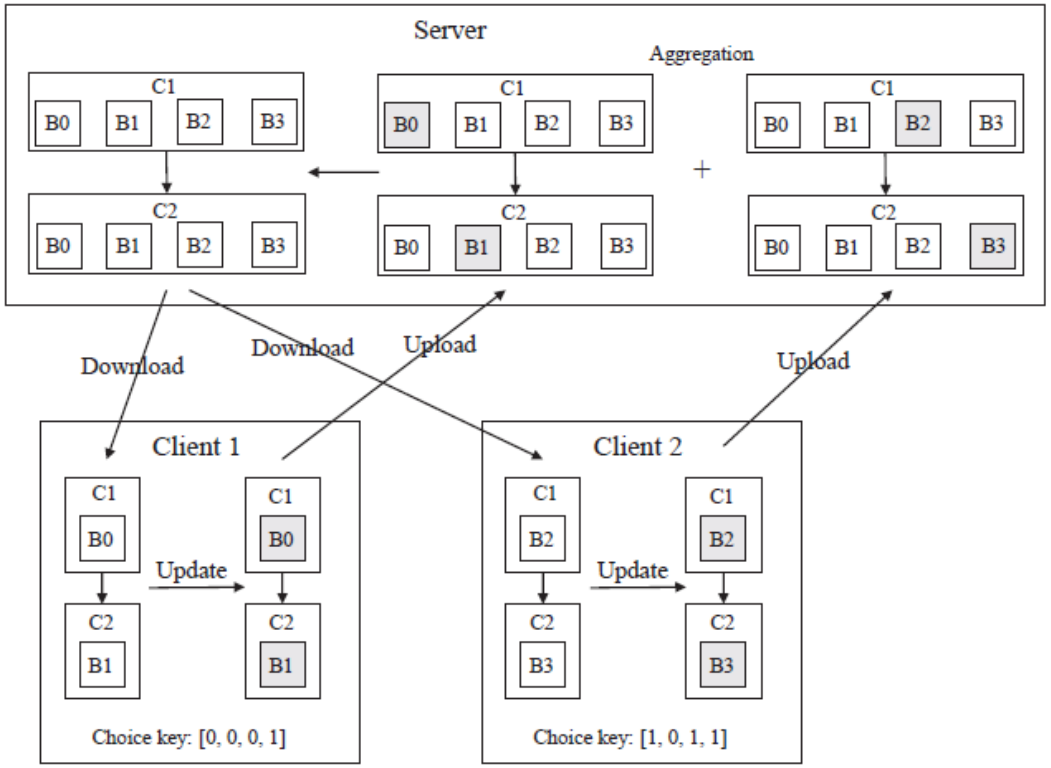
Client sampling



Federated Neural Architecture Search

UNIVERSITY OF SURREY

Master Model Aggregation



- Model filling: Different clients may have sub-networks with different sampled branches. Thus, the “unused paths” need to be filled with the model paths in the server to let every uploaded models have the same structure for afterwards model aggregation.

- Aggregation: Do weighted averaging like FedAvg algorithm after model filling

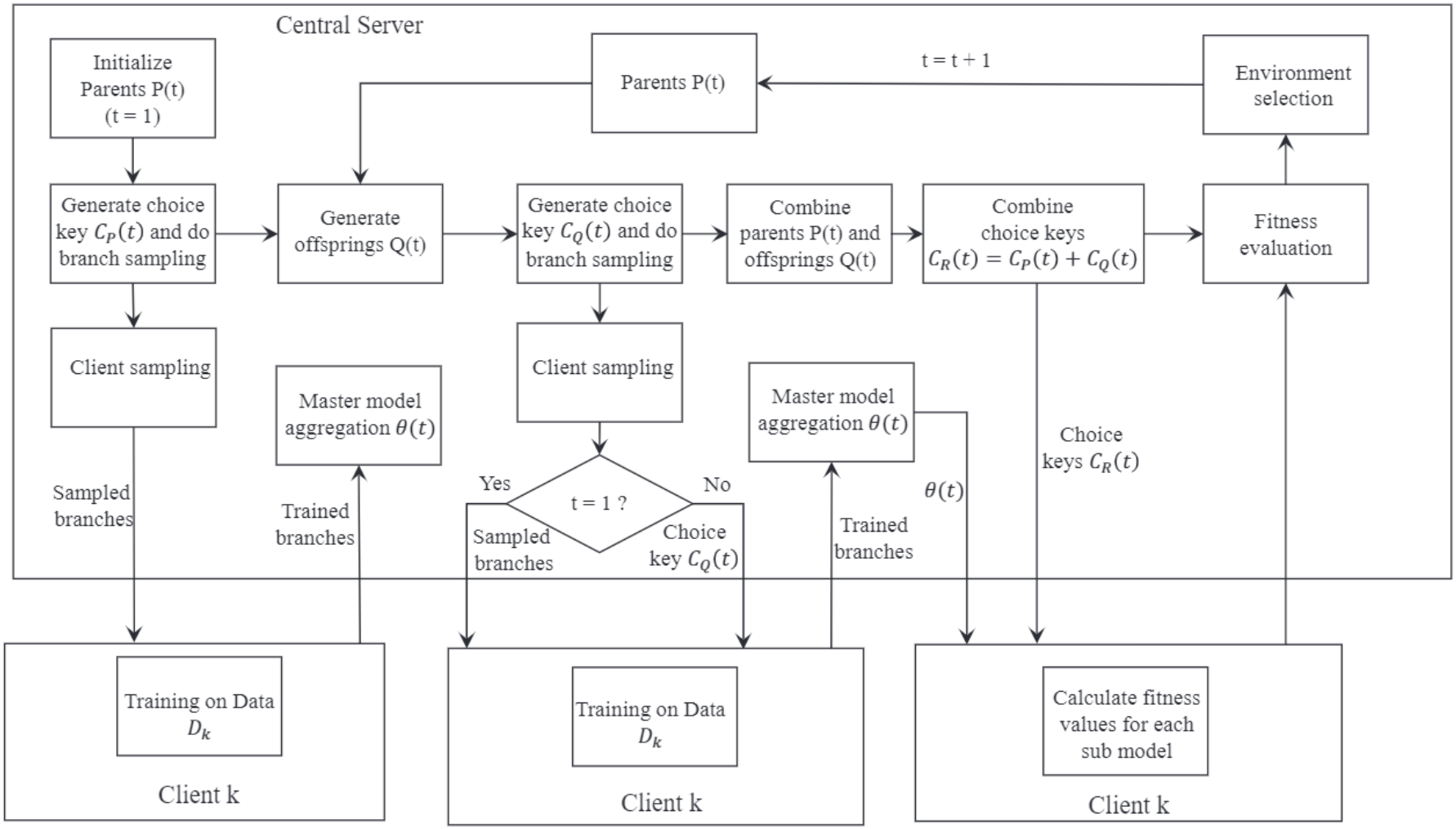
Federated Neural Architecture Search

UNIVERSITY OF SURREY

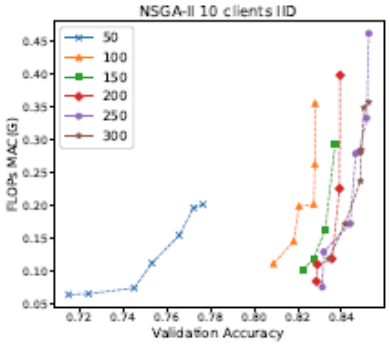
Objectives:

- The model performance (validation accuracy of each sub network)
- The model FLOPs of each sub network
- The number of model parameters of each sub network

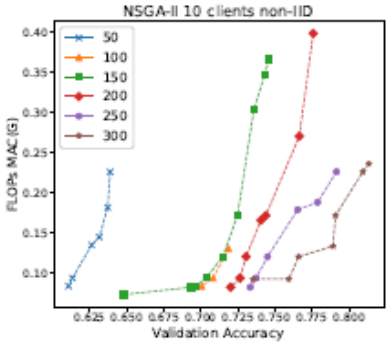
Federated Neural Architecture Search



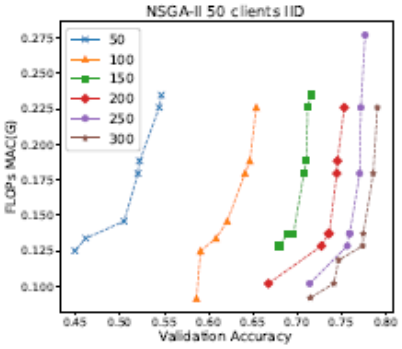
Federated Neural Architecture Search



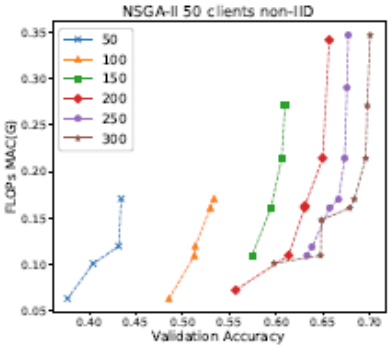
(a) 10 clients, IID data



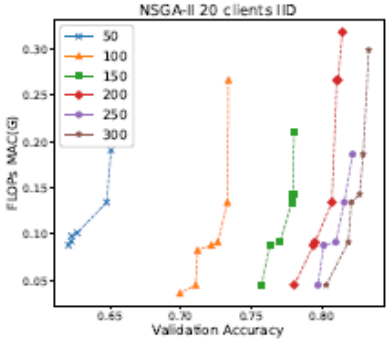
(b) 10 clients, non-IID data



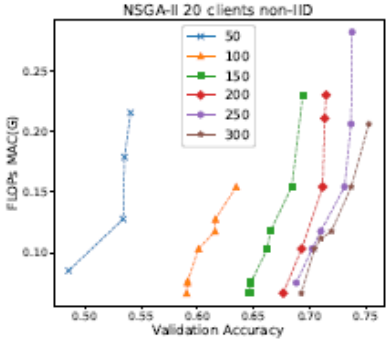
(e) 50 clients, IID data



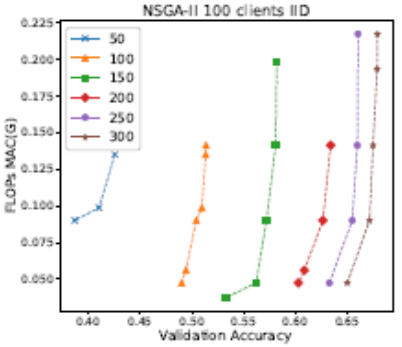
(f) 50 clients, non-IID data



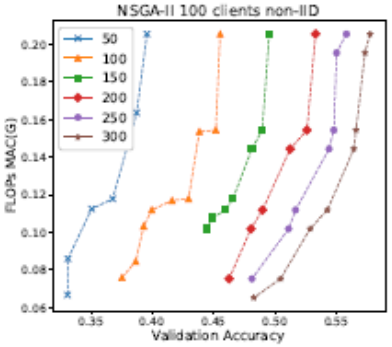
(c) 20 clients, IID data



(d) 20 clients, non-IID data

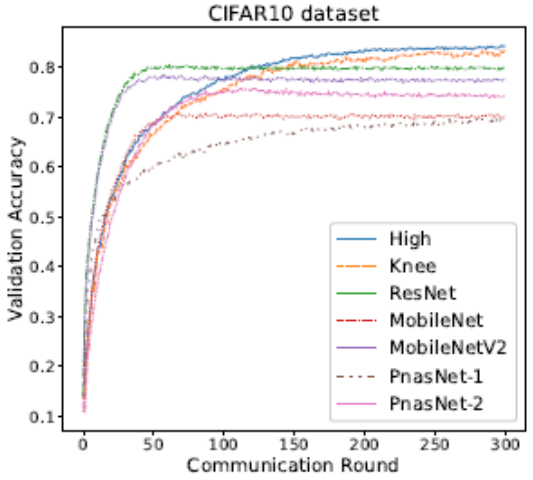


(g) 100 clients, IID data

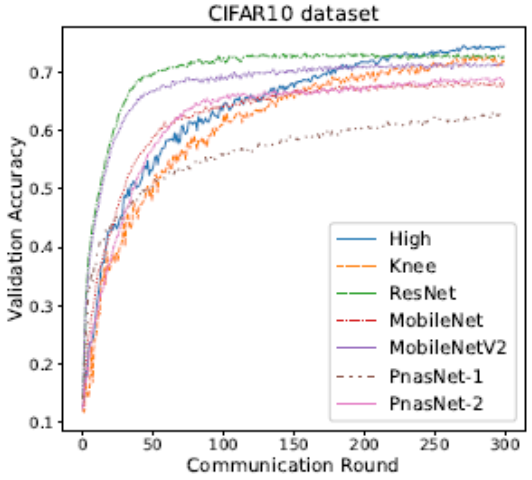


(h) 100 clients, non-IID data

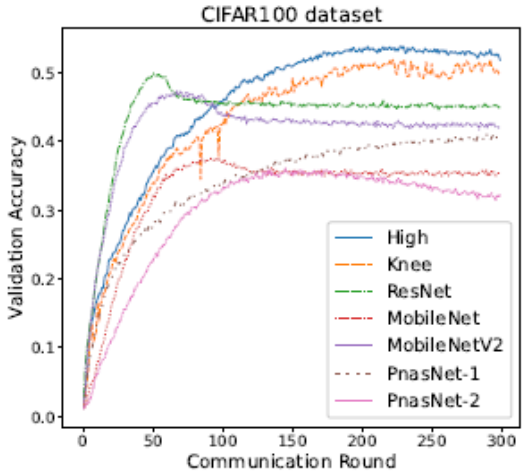
Federated Neural Architecture Search



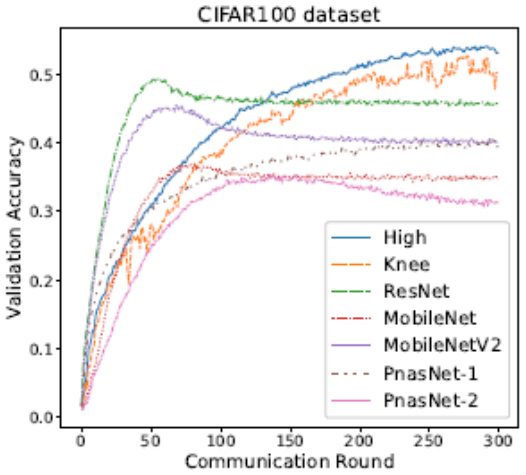
(a) Validation Accuracy, IID data



(b) Validation Accuracy, non-IID data

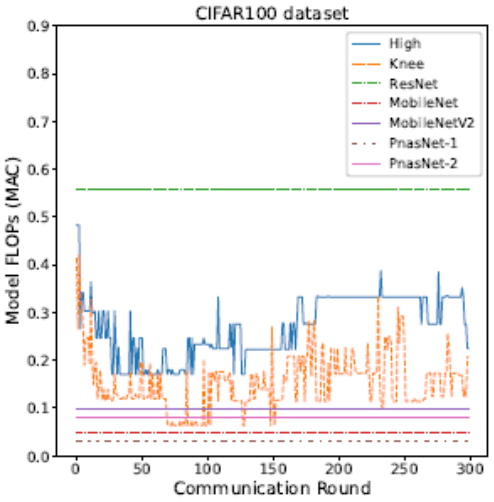


(a) Validation Accuracy, IID data

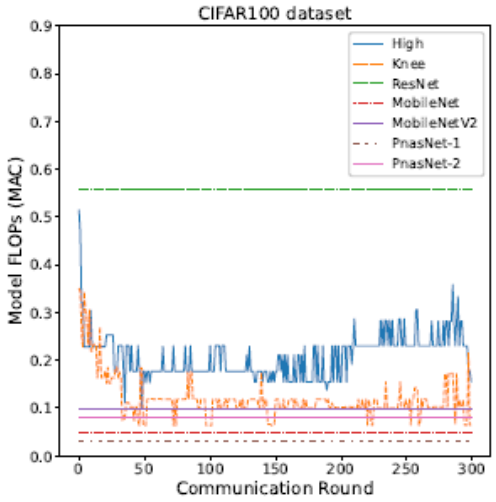


(b) Validation Accuracy, non-IID data

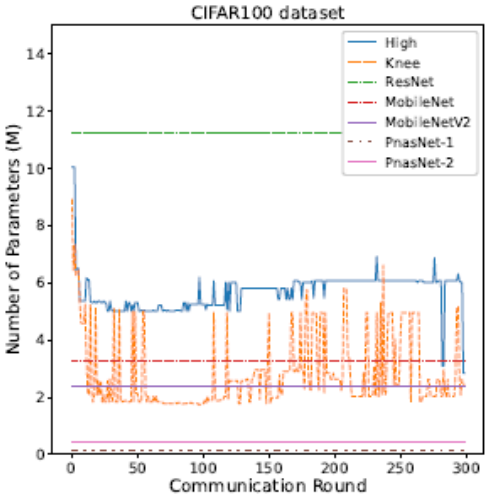
Federated Neural Architecture Search



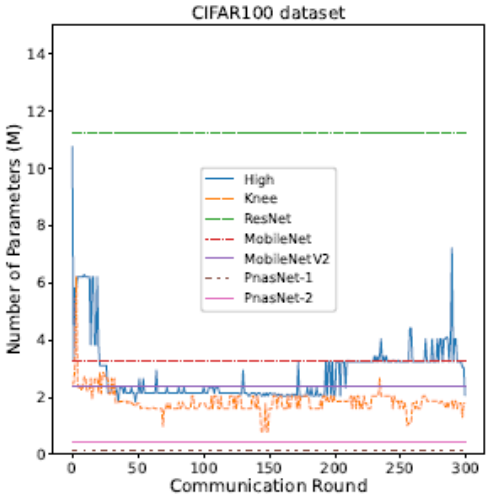
(c) FLOPs, IID data



(d) FLOPs, non-IID data



(e) Num of Params, IID data



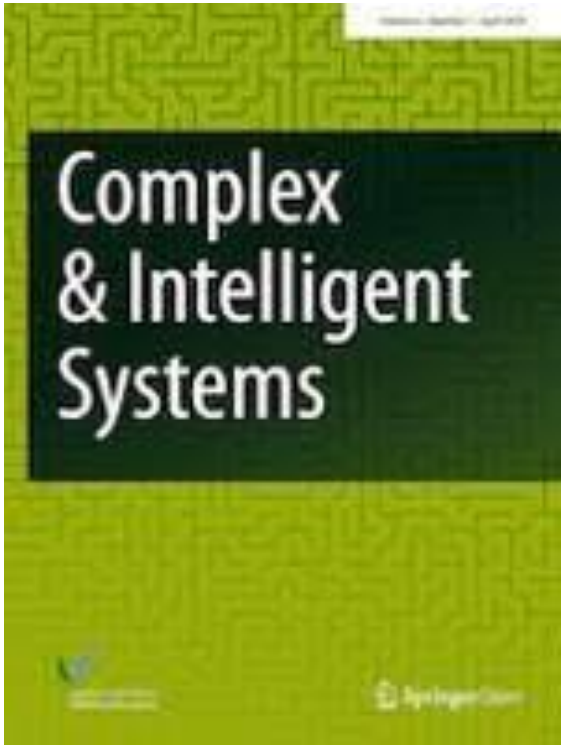
(f) Num of Params, non-IID data

Summary and Future Work

- Summary
  - An evolutionary multi-objective FL is also proposed, which can significantly reduce communication cost and improve learning performance
  - A real-time evolutionary federated neural architecture search method is proposed
  - Evolutionary multi-objective learning provides a powerful approach to handling multiple (conflicting) objectives

- Future work
  - More secure federated learning consider adversarial attacks and communication loss
  - Vertical federated learning

H. Zhu, R. Wang, Y. Jin, K. Liang, and J. Ning. Distributed additive encryption and quantization for privacy preserving federated deep learning. https://arxiv.org/abs/2011.12623



Editor-in-Chief: Yaochu Jin

Impact factor (2019): **3.791**

- **Complexity**
  - Complex evolutionary and adaptive Systems
  - Emergent properties and behavior in complex

- **Systems**
  - Self-organizing collective systems
  - Biological and social inspirations in problem solving
  - Systems Science and Engineering

  Intelligent Data Analytics
  - Data mining and knowledge discovery
  - Machine learning
  - Pattern recognition
  - Big Data Analytics and data science
  - Data-driven problem solving

- **Computational Simulation**
  - Knowledge-based Systems
  - Agent based Systems
  - Uncertainty modeling
  - Decision support Systems
  - Brain-like computing
  - Ubiquitous computing
  - Computational visualization and interaction

Acknowledgements: Thanks to my (visiting) PhD students Hangyu Zhu, Jinjin Xu, and Yang Cheng who contributed to the work presented in the talk.

Many thanks for your attention!