Towards Edge ML: Principles, Current State, and Perspectives

Hakim Hacid

Technology Innovation Institute, UAE

January 10, 2023

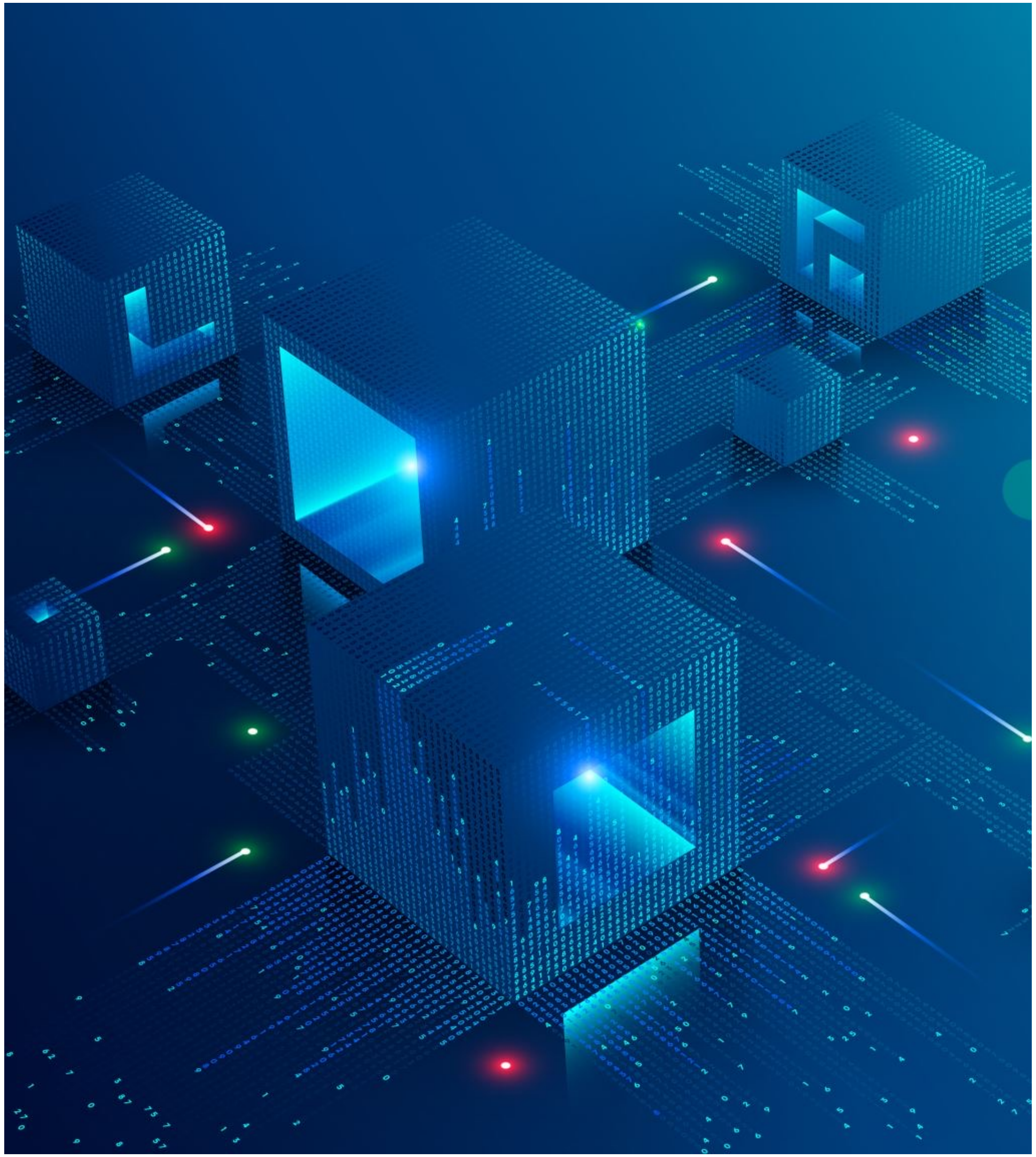
Workshop FL-Day
Decentralized Federated Learning: Approaches and Challenges

Technology Innovation Institute (TII)

TII is the applied research arm of the Advanced Technology Research Council (ATRC).

Designed to push the frontiers of knowledge and create a culture, TII is a **world-class R&D Institute** focusing on applied research and advanced technology via dedicated **Research Centers**.

Working with exceptional talent, universities, research institutions and industry partners from around the world, TII brings together an intellectual community and contributes to the **UAE's growing** R&D ecosystem and knowledge-based economy.



TII

Research centers

- Advanced Materials
- Autonomous Robotics
- Biotechnology
- Cryptography
- Directed Energy
- Digital Science
- Propulsion and Space
- Quantum
- Renewable and Sustainable Energy
- Secure Systems

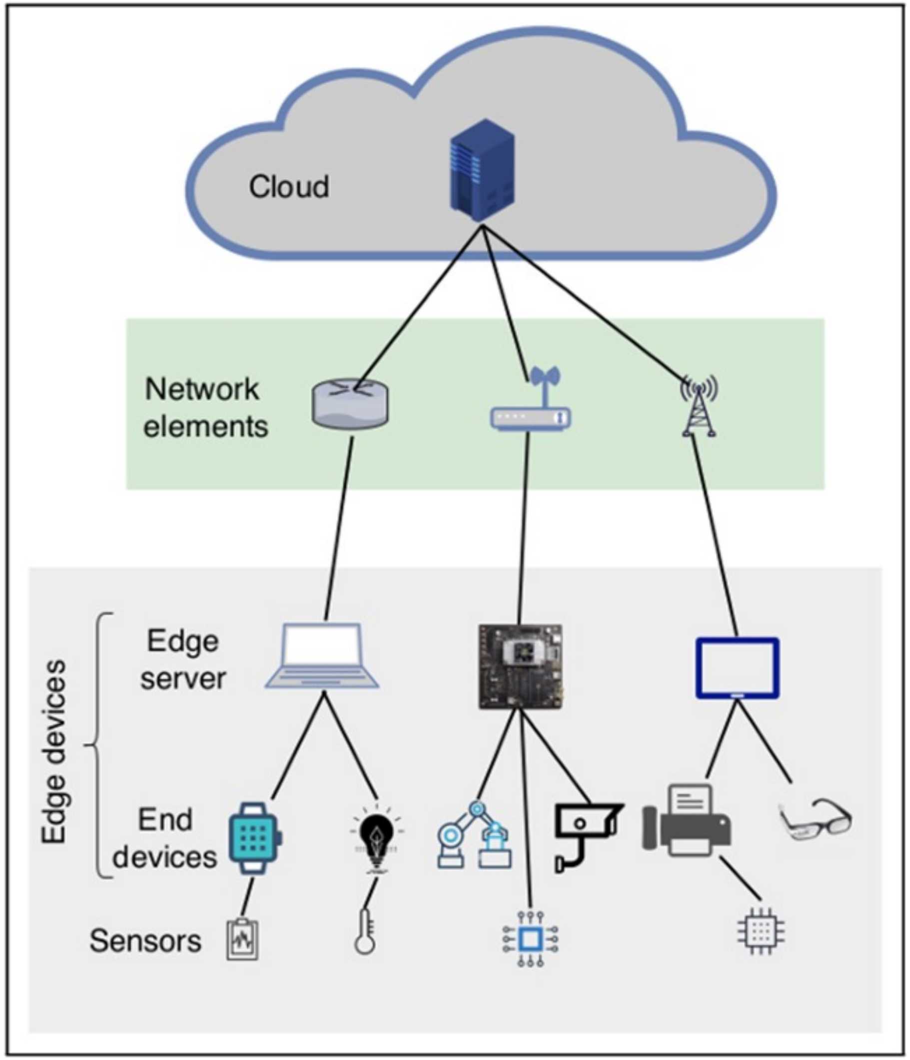
Outline

- Introduction
- Edge Machine learning
- Example application: Human activity recognition
- Future research directions
- Conclusion

Introduction

- **Cloud computing**: The on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user.
- **Edge computing**: A set of techniques that bring data collection, analysis, and processing to the edge of the network.
- **Machine Learning**: A set of techniques for understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks.

Introduction



M. G. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, “Machine Learning at the Network Edge: A Survey”, ACM Computing Surveys, vol. 54, no. 8, article no. 170, pp. 1-37

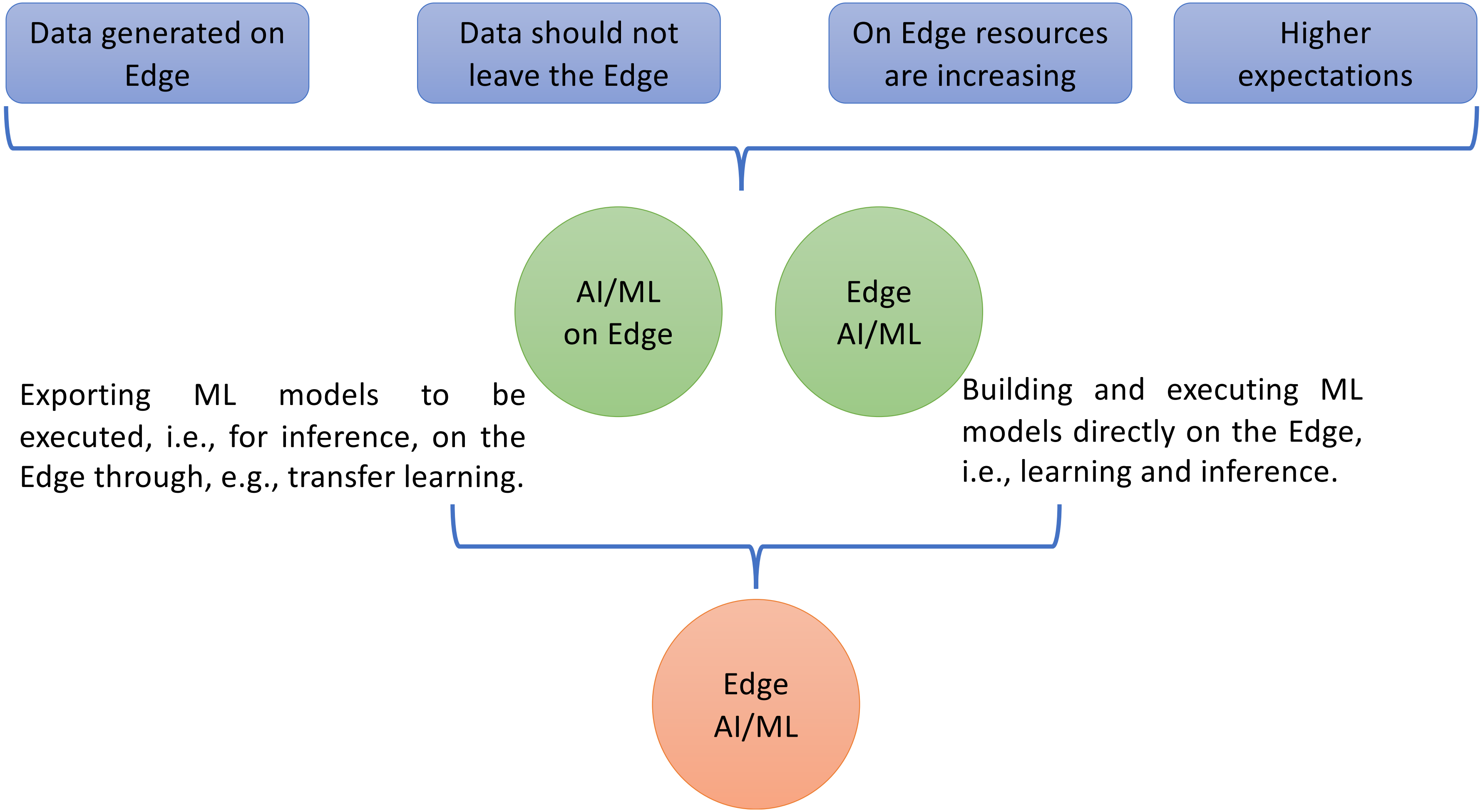
Introduction Motivation

- Small devices increased in penetration and proliferation
- Connectivity has become ubiquitous
- Amount of the collected data is increasing exponentially

- Privacy and security concerns are growing
- High costs or constraints on large data transmission for training purposes.
- Increase in the computation power available at the IoT-scale devices
- Higher demand and expectation on the performance, i.e., latency.

Introduction

Motivation



Introduction

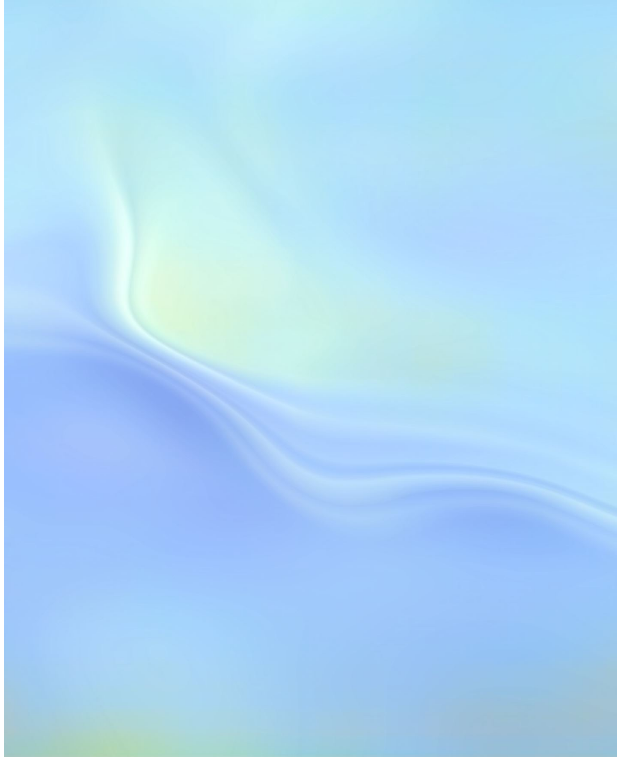
Edge Machine Learning

A combination of	Edge Computing Machine Learning
Aiming at	Building and executing ML models directly on the Edge devices
Operated on	A single or multiple devices
Constrained by	Not sharing data with other devices or clouds and communication
Offering	Comparable quality Better privacy and security Lower latency Offline capabilities

Introduction
Edge Machine Learning

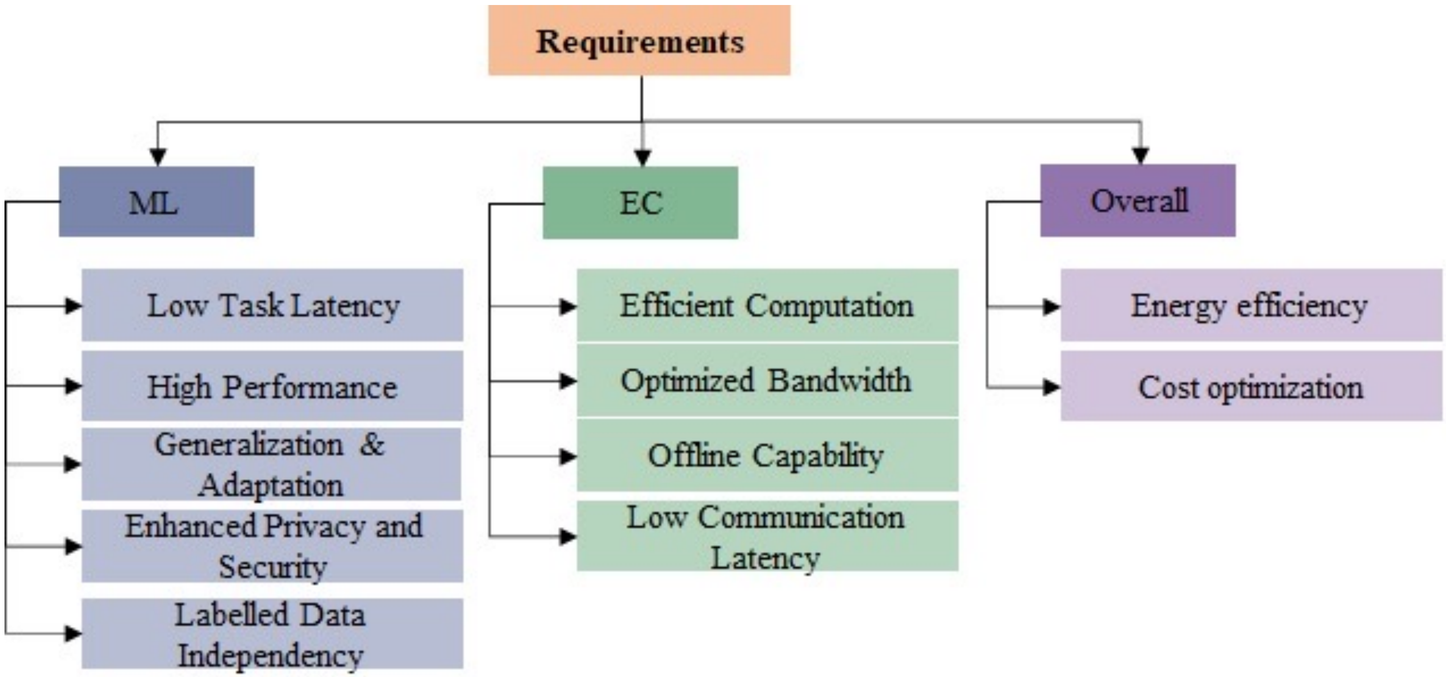
Examples of questions that Edge ML attempts to answer:

- Is there a real need to let the data leave the Edge device?
- How can we allow a training to happen directly on the Edge?
- How to efficiently execute models, i.e., inference, on the Edge?



Introduction

Edge ML Requirements



A Review and a Taxonomy of Edge Machine Learning: Requirements, Paradigms, and Techniques

WENBIN LI, Technology Innovation Institute, United Arab Emirates
HAKIM HACID, Technology Innovation Institute, United Arab Emirates
EBTESAM ALMAZROUEI, Technology Innovation Institute, United Arab Emirates
MEROUANE DEBBAH, Technology Innovation Institute, United Arab Emirates

The union of Edge Computing (EC) and Artificial Intelligence (AI) has brought forward the Edge AI concept to provide intelligent solutions close to end-user environment, for privacy preservation, low latency to real-time performance, as well as resource optimization. Machine Learning (ML), as the most advanced branch of AI in the past few years, has shown encouraging results and applications in the edge environment. Nevertheless, edge powered ML solutions are more complex to realize due to the joint constraints from both edge computing and AI domains, and the corresponding solutions are expected to be efficient and adapted in technologies such as data processing, model compression, distributed inference, and advanced learning paradigms for Edge ML requirements. Despite that a great attention of Edge ML is gained in both academic and industrial communities, we noticed the lack of a complete survey on existing Edge ML technologies to provide a common understanding of this concept. To tackle this, this paper aims at providing a comprehensive taxonomy and a systematic review of Edge ML techniques: we start by identifying the Edge ML requirements driven by the joint constraints. We then survey more than twenty paradigms and techniques along with their representative work, covering two main parts: edge inference, and edge learning. In particular, we analyze how each technique fits into Edge ML by meeting a subset of the identified requirements. We also summarize Edge ML frameworks and open issues to shed light on future directions for Edge ML.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Machine learning**; *Distributed algorithms*; *Model development and analysis*.

Edge ML landscape

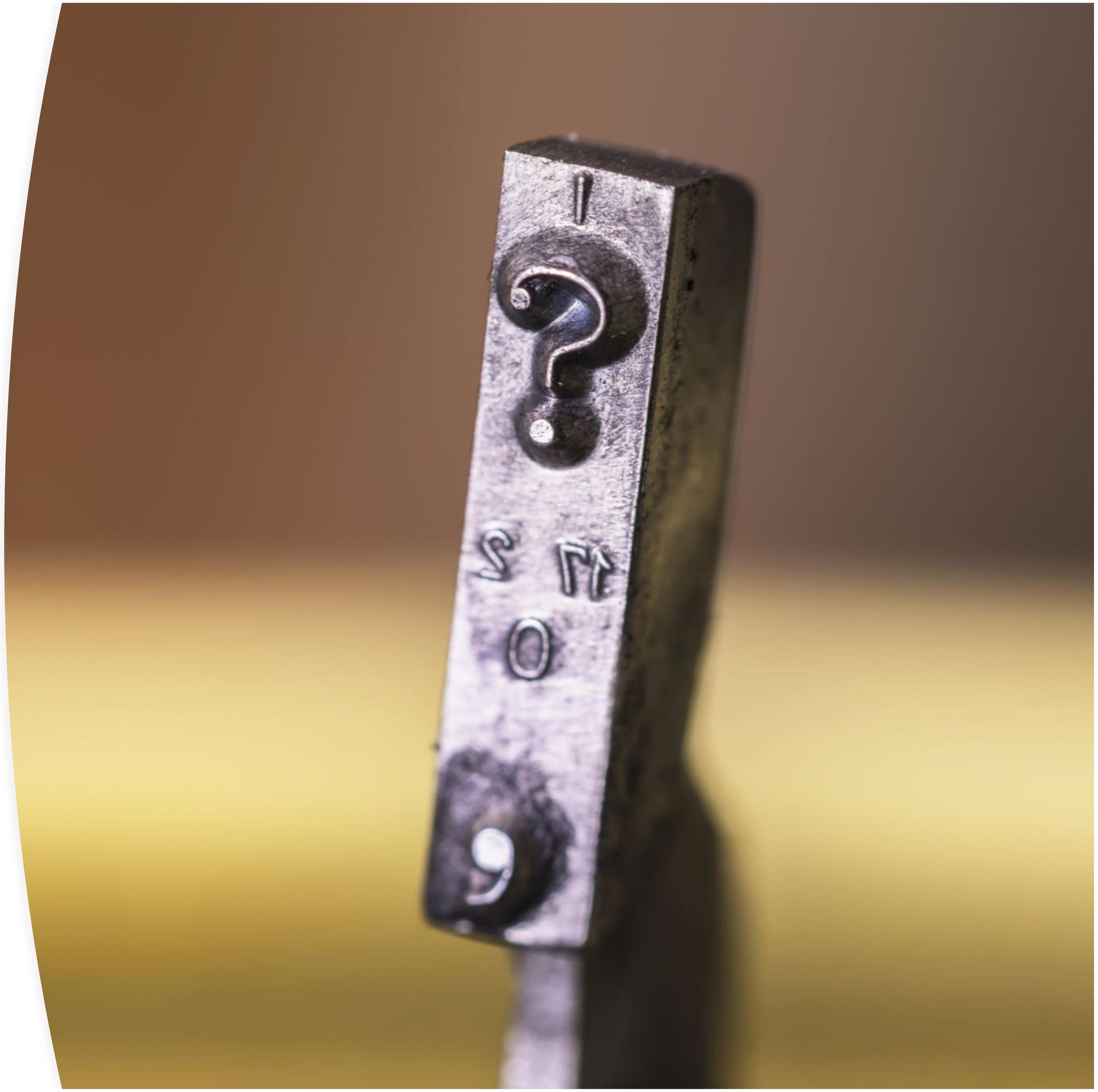
INFERENCE ON EDGE

LEARNING ON EDGE

DATA PRE-PROCESSING ON EDGE

Inference on Edge

- Considers two aspects:
  1. How to use (large) ML models on Edge devices
  2. How to increase the inference efficiency
- Two main categories of methods:
  1. Model compression and approximation
  2. Distributed inference



Inference on Edge
Model Compression and approximation

- Methods that transform ML models into smaller size or approximate models to:
  - Reduce the memory use and the arithmetic operations during the inference
  - Keeping acceptable performances
- Three main sub-categories can be observed here:
  - Quantization
  - Weight Reduction
  - Activation Function Approximation

Inference on Edge

Model Compression and approximation

- Quantization
  - ML model's parameters and activation outputs conversion from one representation to another
  - Usually from Floating Point (FP) format of high precision, e.g., FP64 or FP32, into a low precision format
  - It can apply on both training and inference steps
- Different types of quantization
  - Low Precision Floating Point Representation
  - Fixed-Point Representation
  - Binarization and Terrorization
  - Logarithmic Quantization

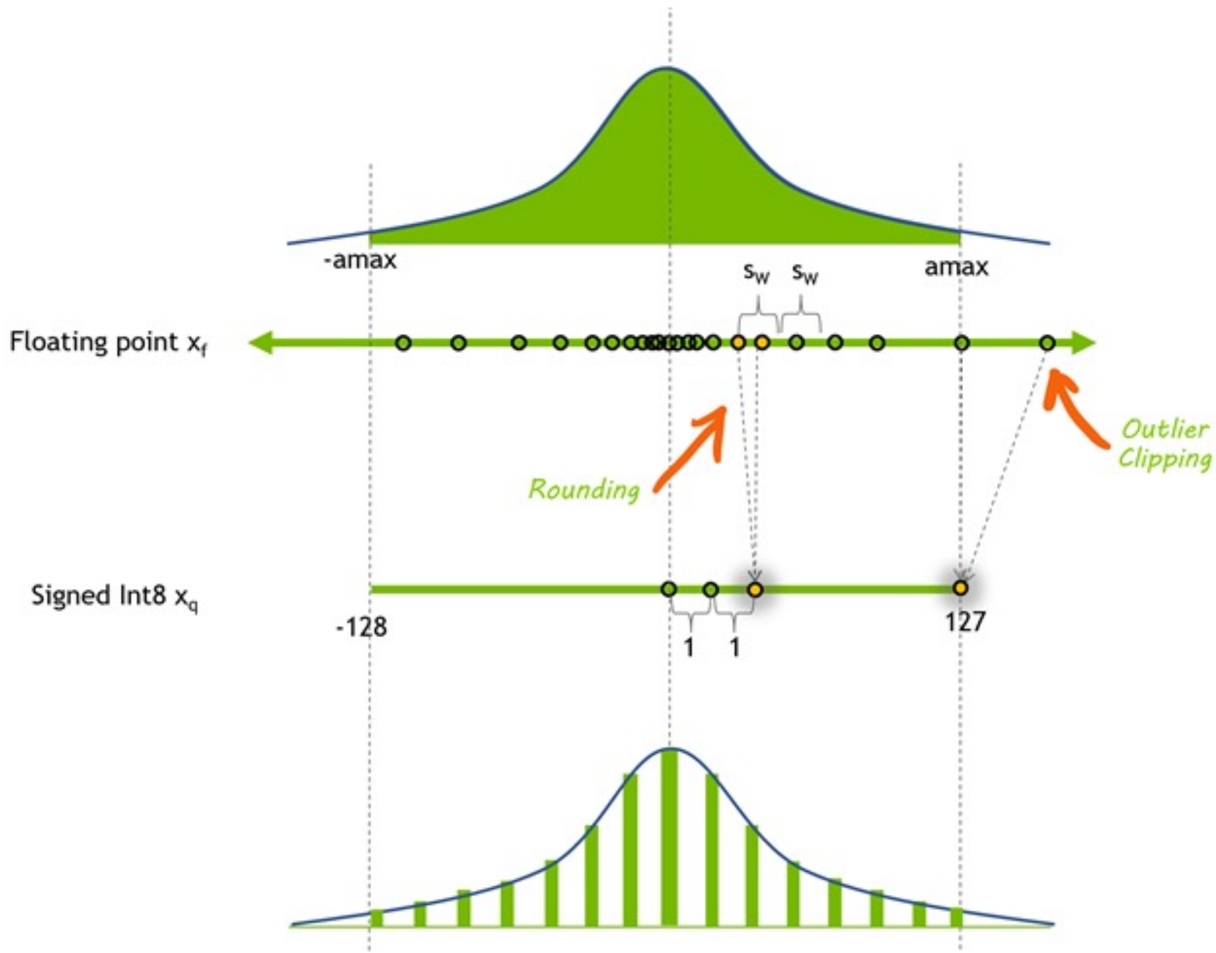


Figure. Illustration of a basic the quantization process (https://developer.nvidia.com/blog/)

Inference on Edge

Model Compression and approximation

- Quantization
  - Useful for ML models on edge devices
  - Decreases the inference task latency by reducing the consumption of computing resources
  - Energy and cost optimization
  - Trade-off between task accuracy and task latency

Work	Quantization	Model size	Quality	Latency
Jacob et al.	8-bit	- 4x	- 1.8%	- 50%
Lee et al.	Logarithmic (4-5 bit)	NA	- 1.5%	NA
Oh et al.	Logarithmic (3 bit)	NA	- 1%	NA

Some illustrative examples of results using quantization

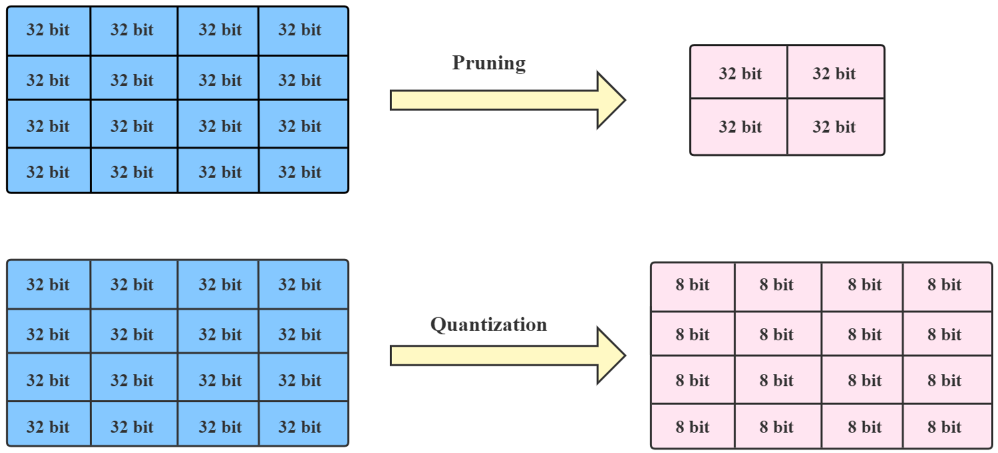
Jacob et al. Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 2704– 2713, dec 2018.

Lee et al. Sugil Lee,Hyeonuk Sim, Jooyeon Choi, and Jongeun Lee. Successive log quantization for cost-efficient neural networks using stochastic computing. Proceedings - Design Automation Conference, (x):2–7, 2019.

Oh et al. Sangyun Oh, Hyeonuk Sim ,Sugil Lee, and Jongeun Lee. Automated Log-Scale Quantizationfor Low-Cost Deep Neural Networks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 742–751, 2021.

Inference on Edge

Model Compression and approximation



- Weight reduction
  - Removal of redundant parameters
  - Two ways of operating it:
    - Pruning
    - parameter approximation.
  - Three categories of methods:
    - **Pruning**: Removing redundant or non-critical weights and/or nodes from models, (Cheng et al.)
    - **Weight Sharing**. Grouping similar model parameters into buckets to be reused (Chu et al.).
    - **Low-rank Factorization**. Decomposing the weight matrix into several low rank matrices by uncovering explicit latent structures (Jaderberg et al.)

Cheng et al. Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A Survey of Model Compression and Acceleration for Deep Neural Networks. oct 2017.
Chu et al. Xiangxiang Chu, Bo Zhang, and Ruijun Xu. FairNAS: Rethinking Evaluation Fairness of Weight Sharing Neural Architecture Search. Proceedings of the IEEE International Conference on Computer Vision, pages 12219–12228, jul. 2021.
Jaderberg et al. Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. BMVC 2014 - Proceedings of the British Machine Vision Conference 2014, may 2014.

Inference on Edge

Model Compression and approximation

Work	Parameters	Quality
Srinivas et al.	- 35%	- 2.2%
Dai et al.	-15.7x to -30.2x	~ - 2%
Gao et al.	- 2x	- 2.54%
Denton et al.	-2.4x to -13.4x	- 0.8%

Some illustrative examples of results using weight reduction

- Weight reduction
  - Removal of redundant parameters
  - Two ways of operating it:
    - Pruning
    - parameter approximation.
  - Three categories of methods:
    - **Pruning**: Removing redundant or non-critical weights and/or nodes from models, (Cheng et al.)
    - **Weight Sharing**. Grouping similar model parameters into buckets to be reused (Chu et al.).
    - **Low-rank Factorization**. Decomposing the weight matrix into several low rank matrices by uncovering explicit latent structures (Jaderberg et al.)

Srinivas et al. Suraj Srinivas and R. Venkatesh Babu. Data-free Parameter Pruning for Deep Neural Networks. pages 31.1–31.12, jul 2015.
Dai et al. Xiaoliang Dai, Hongxu Yin, and Niraj K. Jha. NeST: A Neural Network Synthesis Tool Based on a Grow-and-Prune Paradigm. IEEE Transactions on Computers, 68(10):1487–1497, 2019.
Gao et al. Xitong Gao, Yiren Zhao, Lukasz Dudziak, Robert Mullins, and Xu Cheng-Zhong. Dynamic channel pruning: Feature boosting and suppression. 7th International Conference on Learning Representations, ICLR 2019, oct 2019.
Denton et al. Emily Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. Advances in Neural Information Processing Systems, 2(January):1269– 1277, apr 2014.

Inference on Edge

Model Compression and approximation

- Weight reduction
  - Reduces the ML model size by removing uncritical parameters
  - ML models use less memory and require fewer arithmetic operations
  - Reduce the task latency with less workload
  - Improve the computational resource efficiency
  - Optimized energy consumption and cost

Inference on Edge

Distributed Inference

- Divides ML models into different partitions
- Performs the inference in a collaborative manner
- Involving different Edge resources in a distributed manner
- Three ways of looking at this:
  - Local processors in the same edge device (Lane et al.)
  - Interconnected edge devices (Zhao et al.)
  - Edge devices combined with cloud servers (Li et al.)
- Challenge:
  - Identify the partition points of the models
  - Measure data exchanges between layers to balance the usage of local computational resources and bandwidth among distributed resources.

Work	Inference time	Memory	Accuracy
Du et al.	- 3.21x	- 65.3%	+ 1.29%
Hemmat et al.	- 17x	NA	- 0.5%

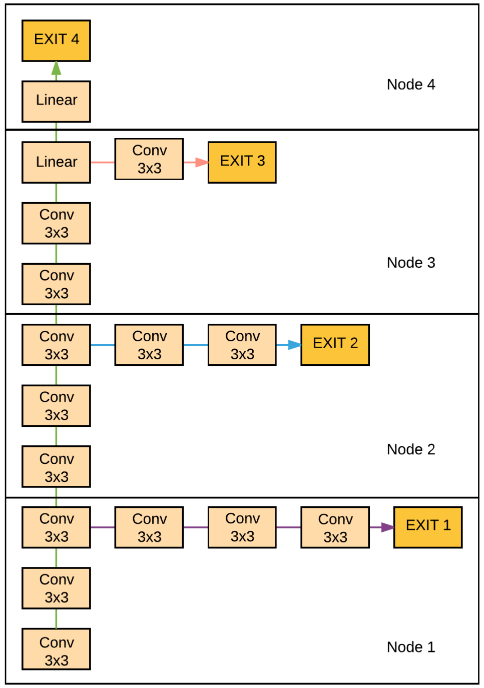
Some illustrative examples of results using Distributed Inference

Zhao et al. Zhuoran Zhao, Kamyar Mirzazad Barijough, and Andreas Gerstlauer. DeepThings: Distributed adaptive deep learning inference on resource-constrained IoT edge clusters. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 37(11):2348–2359, nov 2018.
Lane et al. Nicholas D. Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, LeiJiao, Lorena Qendro, and Fahim Kawsar. DeepX: A Software Accelerator for Low-Power Deep Learning Inference on Mobile Devices. 2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN 2016 - Proceedings, apr 2016.
Li et al. He Li, Kaoru Ota, and Mianxiong Dong. Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. IEEE Network, 32(1):96–101, jan 2018.
Du et al. Jiangsu Du, Xin Zhu, Minghua Shen, Yunfei Du, Yutong Lu, Nong Xiao, and Xiangke Liao. Model Parallelism Optimization for Distributed Inference Via Decoupled CNN Structure. IEEE Transactions on Parallel and Distributed Systems, 32(7):1665–1676, jul 2021.
Hemmat et al. Maedeh Hemmat, Azadeh Davoodi, and Yu Hen Hu. EdgenAI: Distributed Inference with Local Edge Devices and Minimal Latency. Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC, 2022- Janua:544–549, 2022.

Inference on Edge

Other Inference Acceleration techniques

- Knowledge Distillation
  - A neural network is trained on the output of another network along with the original targets in order to transfer knowledge between the ML model architectures
- Activation approximation
  - Replaces non-linear activation functions (e.g., Sigmoid) with less computational expensive functions (e.g., ReLU)
  - Simplify the calculation or convert the computational expensive calculation to series of lookup tables.
- Early Exit of Inference (EEoI)
  - Powered by a deep network architecture augmented with additional side branch classifiers (Teerapittayanon et al.)
  - Enables early decision on some cases when it is high enough
- Inference Cache
  - Saves models or models' inference results to facilitate future inferences of similar interest
  - ML tasks requested by nearby users within the coverage of an edge node may exhibit spatio-temporal locality
  - Reduces task latency on continuous inference tasks or task batch



High level illustration of EEoI mechanism

Work	Inference time
Teerapittayanon et al.	- 2-6x
Zhou et al.	- 2.42x

Teerapittayanon et al. Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. Proceedings - International Conference on Pattern Recognition, 0:2464–2469, sep 2016.

Zhou et al. Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. BERT loses patience: Fast and robust inference with early exit. Advances in Neural Information Processing Systems, 2020.

Inference on Edge

Reflection!

- A veriety of methods
- Promising results
- Lack of real “public” devices implementations
- Lack of automation
- No killer app yet!
- … More work is needed

Learning on Edge

Build ML models directly on Edge devices

Relies on locally stored data

Different approaches, including:

- Distributed learning
- Transfer learning
- Meta learning
- Self-supervised learning

Learning on Edge

Distributed Learning

- Build the leaning model directly on the edge devices
- Divides the model training workload onto the edge nodes
- Jointly train models with a cloud server by taking advantage of individual edge computational resources
- Client server approach
  - Clients transmit locally updated model parameters or locally calculated outputs to the aggregation servers
  - The aggregation server constructs the global model with all shared local updates
- Peer-to-Peer approach:
  - The model is built incrementally along with the participating edge nodes together

Learning on Edge
Distributed Learning

- One can distinguish three approaches:
  - Cloud Enabled DL
    - Aggregation is performed at cloud server side
    - Good performance
    - Large communication burden
  - Edge Enabled DL
    - Aggregation is performed at the Edge server side
    - Less communication burden
    - Offline mode possible
    - Limitations in terms of devices that can be supported
  - Hierarchical DL
    - Combination of both previous approaches
    - Intends to support issues like Non-IID, and unbalanced classes
    - Good for device heterogeneity

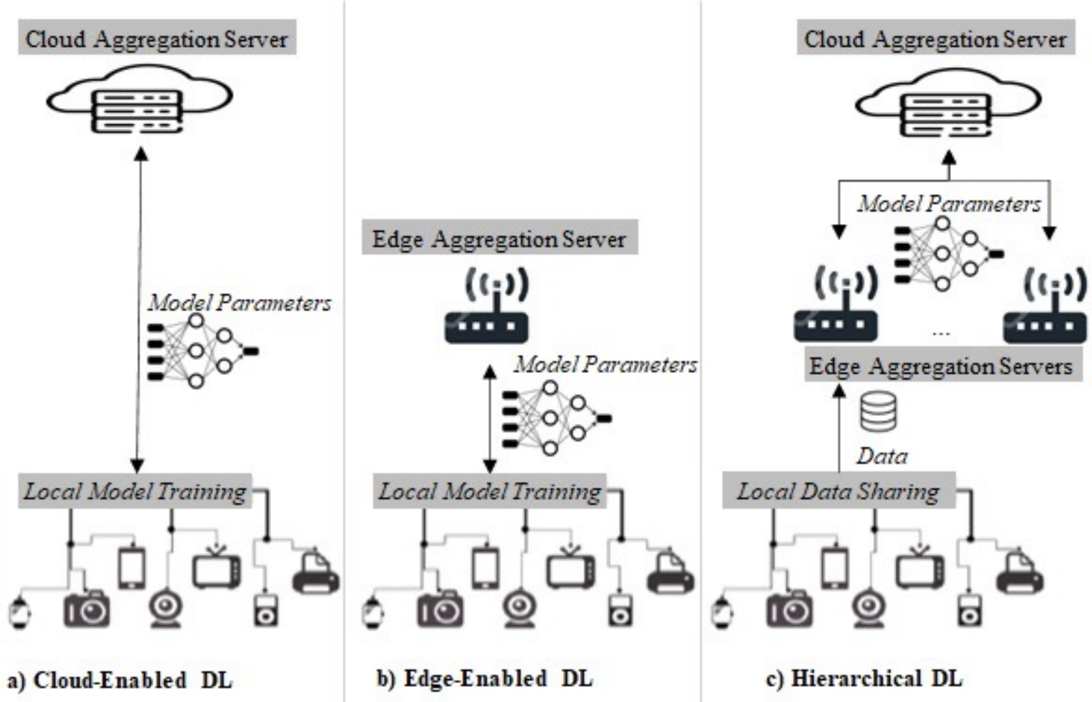


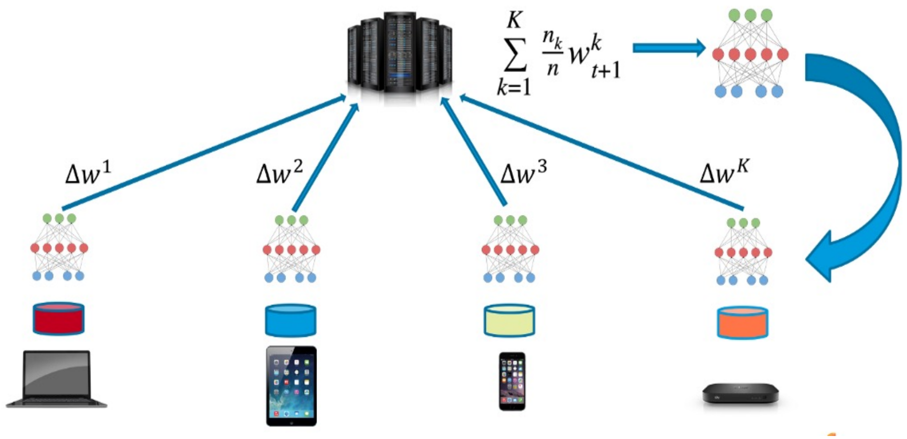
Fig. Different approaches for distributed learning (Abreha et al., Wang et al.)

Abreha et al. Haftay Gebreslasie Abreha, Mohammad Hayajneh, and Mohamed Adel Serhani. Federated Learning in Edge Comput- ing: A Systematic Survey. Sensors, 22(2), 2022.
Wang et al. Xiaofei Wang, Yiwen Han, Victor C.M. Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. IEEE Communications Surveys and Tutorials, 22(2):869– 904, apr 2020.

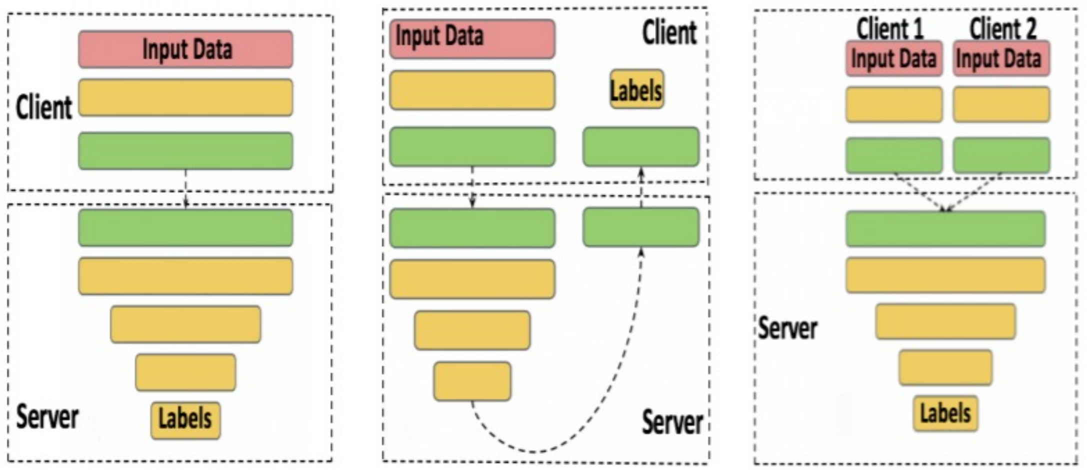
Learning on Edge

Distributed Learning

- Federated Learning (McMahan et al.)



- Split Learning (Gupta et al.)



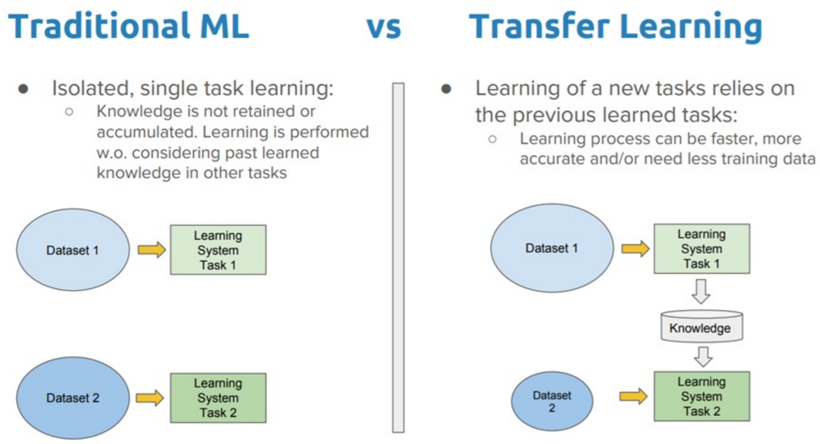
(a) Simple vanilla split learning (b) Split learning without label sharing (c) Split learning for vertically partitioned data

McMahan et al. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273-1282). PMLR.

Gupta et al. Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. Journal of Network and Computer Applications, 116:1–8, oct 2018.

Learning on Edge

Transfer Learning



(Image credit: Dipanjar Sarkar)

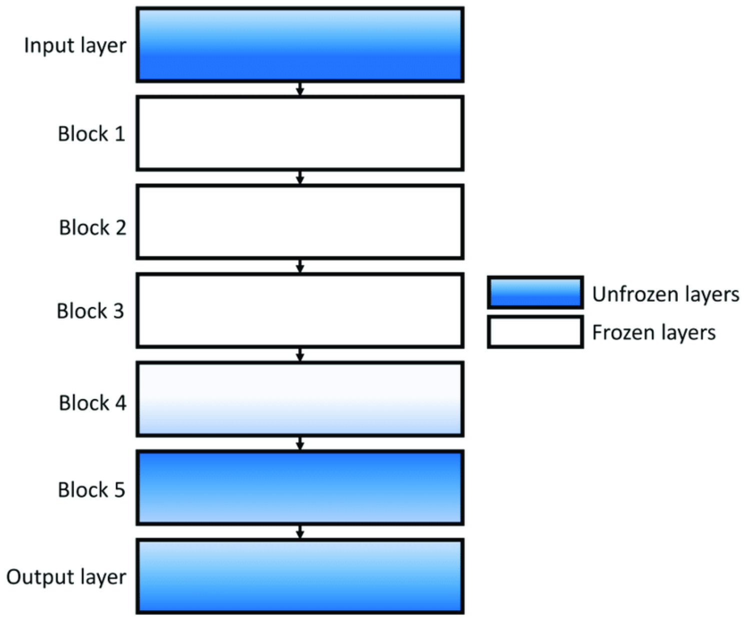
- Human ability to transfer knowledge
- Teacher-student type of processes
- Creating high-performance models on a target domain (student) by transferring the knowledge from models of a different but correlated source domain (Teacher).
- Can be operated on three levels:
  - Data Distribution: Domain Adaptation (Zhang et al.)
  - Feature Space (Pan et al.)
  - Learning Task Space: Domain Generalization, i.e., correlated tasks, (Zhou et al.)

Zhang et al. Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I. Jordan. Bridging theory and algorithm for domain adaptation. 36th International Conference on Machine Learning, ICML 2019, 2019-June:12805–12823, apr 2019.
Pan et al. Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2010.
Zhou et al. Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain Generalization: A Survey. mar 2021.

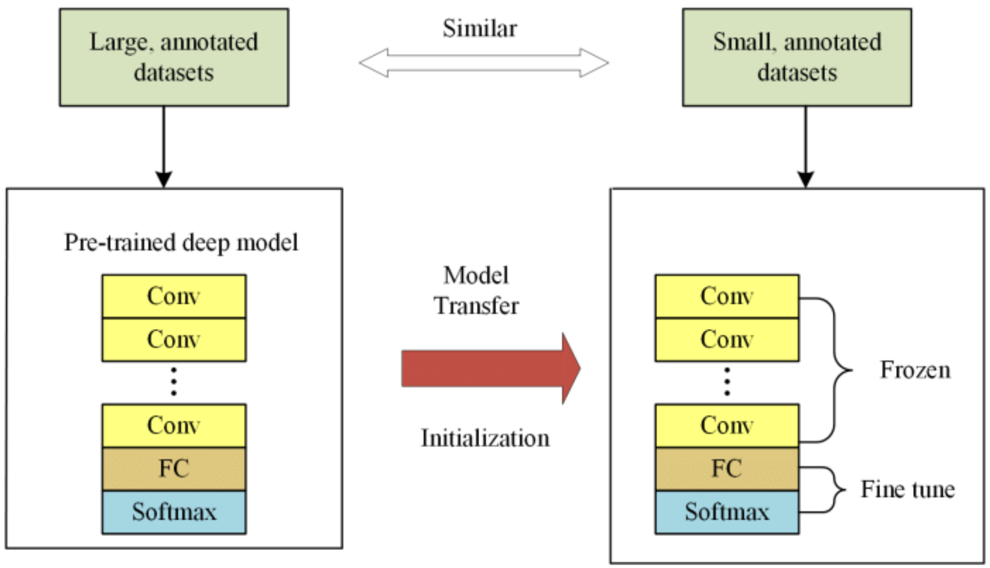
Learning on Edge

Transfer Learning

- Layer Freezing



- Model Tunning



Edge ML

Summary

Learning on Edge
Summary

Edge ML Techniques			Edge ML Requirements									
			ML					edge			Overall	
			Low Task Latency	High Performance	Generalization & Adaptation	Enhanced Privacy and Security	Labelled Data Independency	Computational Efficiency	Optimized Bandwidth	Offline Capability	Energy Efficiency	Cost Optimization
Edge Inference	Model Compression & Approximation	Quantization	+	-	/	/	/	+	/	/	+	+
		Weight Reduction	+	-	/	/	/	+	/	/	+	+
		Knowledge Distillation	+	-	/	/	/	+	/	/	+	+
		Activation Function Approximation	+	-	/	/	/	+	/	/	+	+
	Distributed Inference		+	/	/	/	/	-	-	-	-	-
	Other Inference Acceleration	Early Exit	+	-	/	/	/	+	/	/	+	+
		Inference Cache	+	-	/	/	/	+	/	/	+	+
		Model-Specific Inference Acceleration	+	-	/	/	/	+	/	/	+	+
Edge Learning	Distributed Learning	Federated Learning	+	-	/	+	/	-	+	-	+	+
		Split Learning	+	/	/	+	/	-	+	-	+	+
	Transfer Learning		+	*	/	/	+	+	/	/	+	+
	Meta-Learning		+	/	+	/	+	*	/	/	*	*
	Self-Supervised Learning		+	+	+	/	+	+	/	/	+	+
	Other Learning Paradigms	Multi-Task Learning	+	+	/	/	/	+	/	/	+	+
		Instance-based Learning	*	*	/	/	+	*	/	/	*	*
		Weakly Supervised Learning	+	/	/	/	+	/	/	/	+	+
		Incremental Learning	-	+	+	/	/	-	-	/	+	+

Example application: Human activity recognition

Learn how to recognize human physical activities directly on Edge

Some Practical Insights on Incremental Learning of New Human Physical Activity on the Edge

George Arvanitakis
Technology Innovation Institute
Abu Dhabi, UAE
George.Arvanitakis@tii.ae

Mthandazo Ndhlovu
Technology Innovation Institute
Abu Dhabi, UAE
Mthandazo.Ndhlovu@tii.ae

Jingwei Zuo
Technology Innovation Institute
Abu Dhabi, UAE
Jingwei.Zuo@tii.ae

Hakim Hacid
Technology Innovation Institute
Abu Dhabi, UAE
Hakim.Hacid@tii.ae

MAGNETO: Edge AI for Human Activity Recognition with New Activities Integration On the Fly

George Arvanitakis
Technology Innovation Institute
Abu Dhabi, UAE

Mthandazo Ndhlovu
Technology Innovation Institute
Abu Dhabi, UAE

Jingwei Zuo
Technology Innovation Institute
Abu Dhabi, UAE

Hakim Hacid
Technology Innovation Institute
Abu Dhabi, UAE

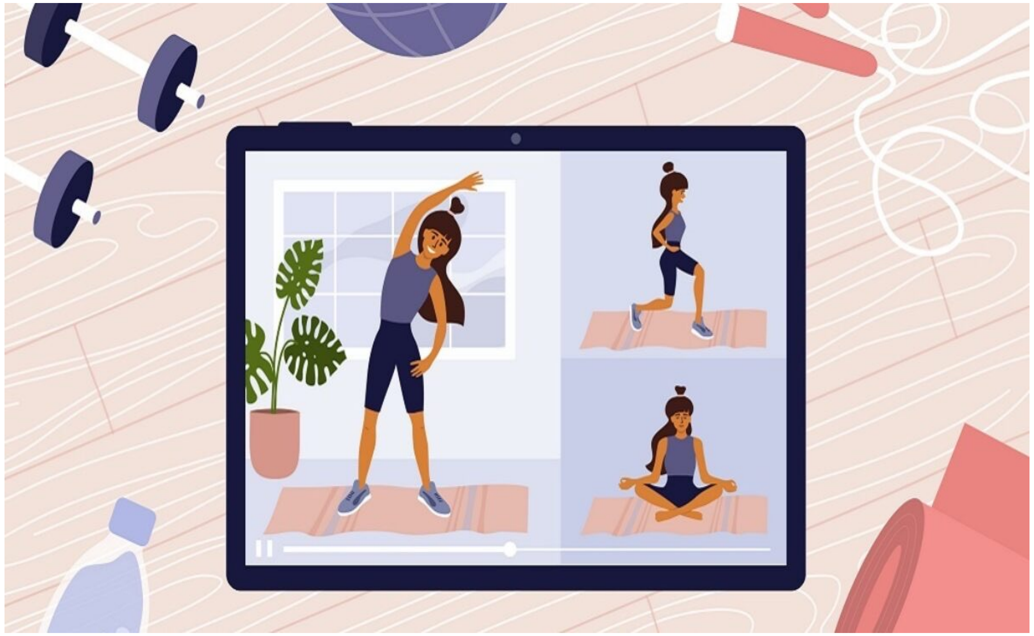
On Handling Catastrophic Forgetting for Incremental Learning of Human Physical Activity on the Edge

Jingwei Zuo
Technology Innovation Institute
Abu Dhabi, UAE
jingwei.zuo@tii.ae

George Arvanitakis
Technology Innovation Institute
Abu Dhabi, UAE
george.arvanitakis@tii.ae

Hakim Hacid
Technology Innovation Institute
Abu Dhabi, UAE
hakim.hacid@tii.ae

Activity Detection and Recognition



Bo Sheng, Oscar Moroni Moosman, Borja Del Pozo-Cruz, Jesus Del Pozo-Cruz, Rosa Maria Alfonso-Rosa, Yanxin Zhang. *A comparison of different machine learning algorithms, types and placements of activity monitors for physical activity classification*. Measurement. Volume 154. 2020, ISSN 0263-2241. https://doi.org/10.1016/j.measurement.2020.107480.

Víctor Labayen, Eduardo Magaña, Daniel Morató, Mikel Izal. *Online classification of user activities using machine learning on network traffic*. Computer Networks. Volume 181. 2020. 107557. ISSN 1389-1286. https://doi.org/10.1016/j.comnet.2020.107557.

Activity Recognition via Smart Devices

Devices

smartphone

smartwatch

UAV

VR/AR

Sensors

smartphone:
- Acceleration
- Gyroscope
- Geomagnetic
- Ambient Light
- Touch
- Fingerprint
- Pressure
- Camera
- Microphone

smartwatch:
- Acceleration
- Gyroscope
- Geomagnetic
- Touch
- Photoplethysmogram (PPG)
- Microphone
- Pressure

UAV:
- Optical
- Thermal
- Lidar (Light Detection and Ranging)
- Ground Penetrating Radar (GPR
- Lightweight Portable Radiometer (LPR)

...

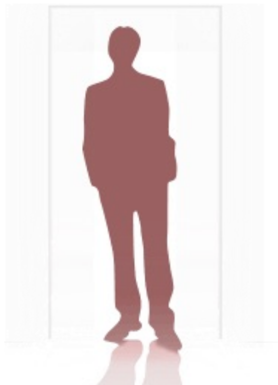
VR/AR:
- Eye tracking
- Head-motion-tracking
- Acceleration
- Gyroscope

Main Applications

- **Commercial purpose (Ads)**
  - when to push a notification?
  - what kind of ads should be promoted?

- **HealthCare**
  - Trace from distance the activities of people in need, e.g., elders and babies.

- **Offensive Security (military)**
  - Estimate and track the activity of a suspect's device. Radiate only when an activity happened.



Activity Recognition via Smart Devices

- Human activity recognition via smart devices
  - Use sensors from ordinary commercial devices to
- Heavily studied topic the last 8 years
- Cloud centric solutions got implemented by almost all tech giants

- How to fully exploit the AI Edge capabilities for activity recognition without sharing personal data?

Samsung Digital Health

SAMSUNG

Developer

Google Cloud

How is it done today and limitations?

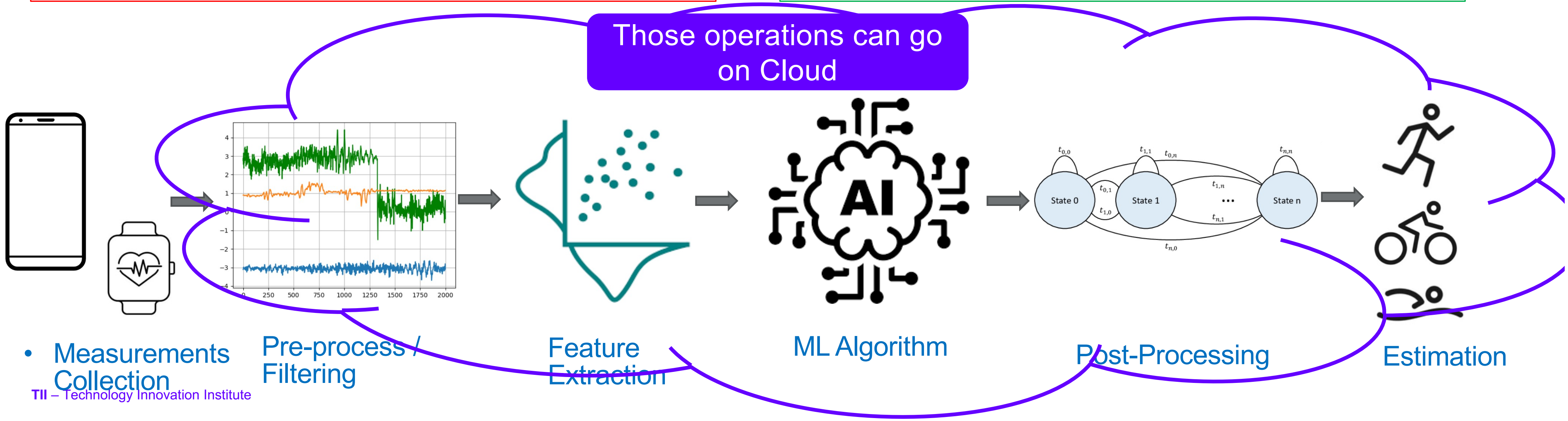
- Centralized, i.e., on the Cloud
- Data transferred to the Cloud
- Inference on the Cloud
- Devices only sense and display

How is it done today and limitations?

Challenges and Opportunities

- Large number of possible activities, often overlapping
- Sensors’ orientation, noise and biases of the hardware
- “Unknown” activities
- Lack of labeled data
- Training of complex ML models are computationally and resources demanding

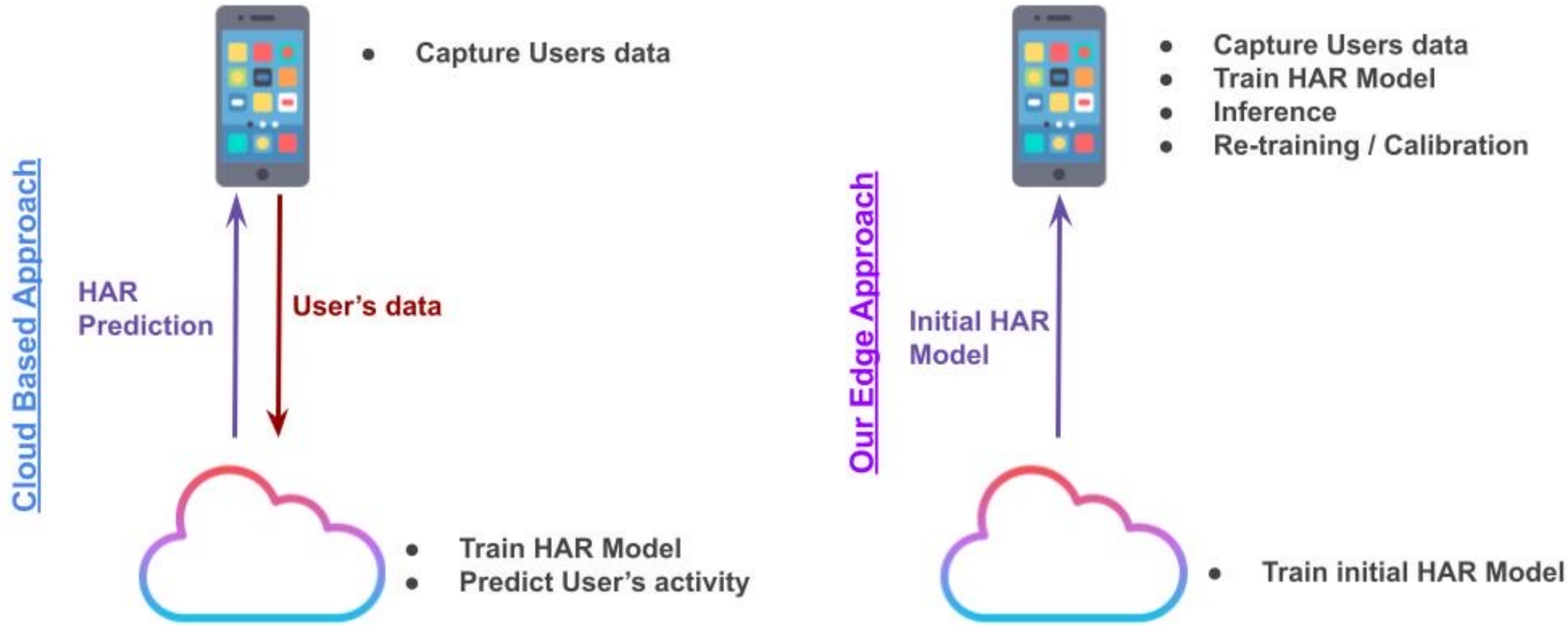
- Nowadays, 8 Billion Smart-phones and Smart-watches capture signals related to human activity
- The same measurements, data curation and data pipelines can be used in multiple problems / functionalities
- Cloud offers scalable resources: computational power, storage and memory, that are mandatory for AI solutions



How is it done today and limitations?

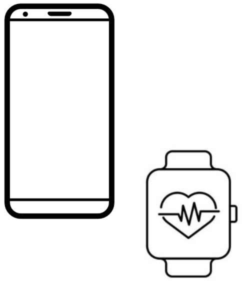
- Personalization Difficulties
- Communication Latency
- Users' data Privacy

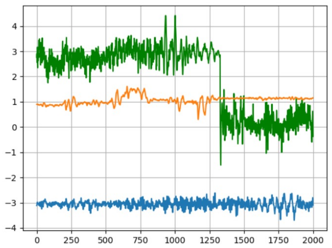
MAGNETO's Edge approach

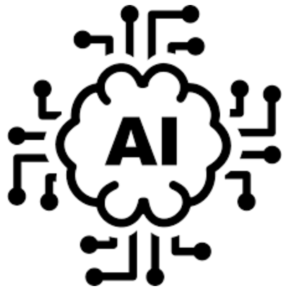


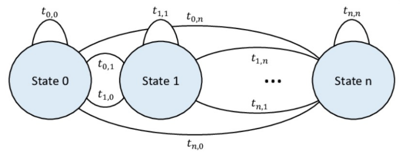
* HAR: Human Activity Recognition

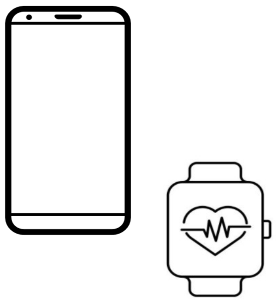
MAGNETO on the Edge





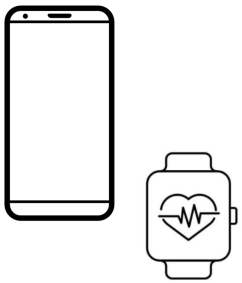


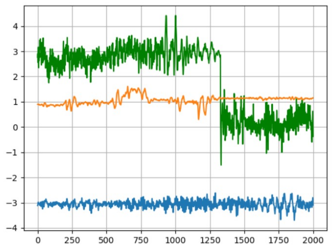


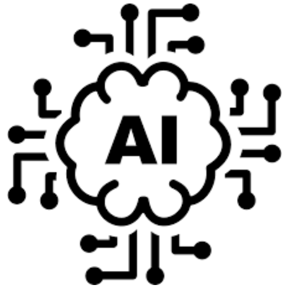


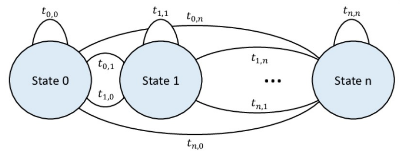
- Smartphones & Smartwatches
- Collected of 25.3 GB of Data
- 11 TII employees
  - 6 RC
  - 8 Nationalities
- ~ 20M samples
- ~ 60 hours of recording

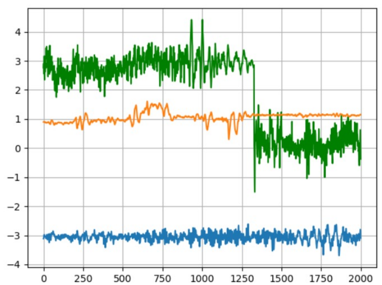
MAGNETO on the Edge







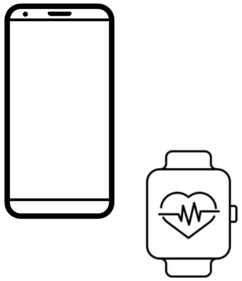


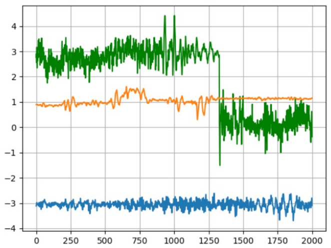


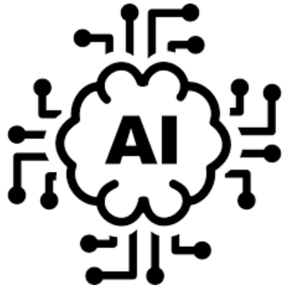
Pre-processing

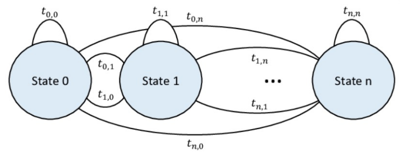
- Scaling
- Outlier detection
- Balance handling / Data Augmentation
- Iterative imputation

MAGNETO on the Edge





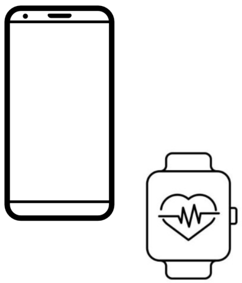


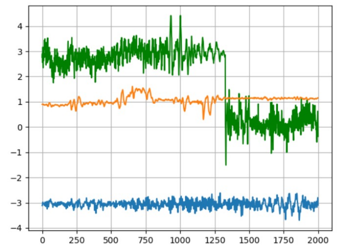


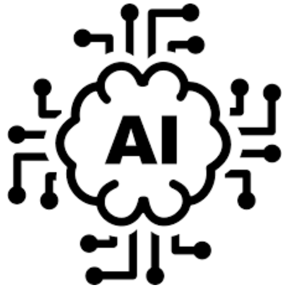
Feature Engineering

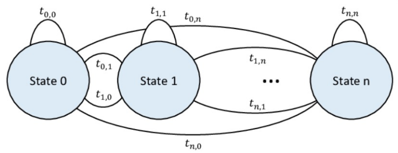
- 1^{st} and 2^{nd} order statistics
- Compute Jerk in 3D Features
- Initial features 21 + 20
- Final number of features 159

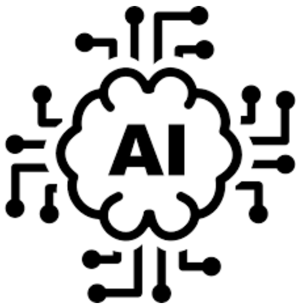
MAGNETO on the Edge







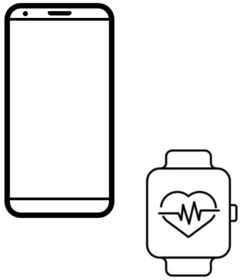


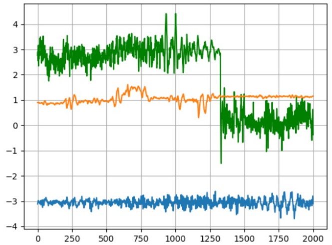


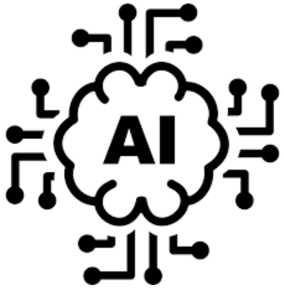
AI Model

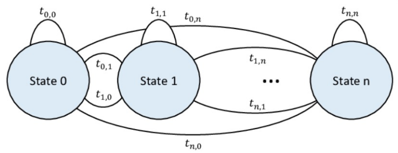
- Siamese Network
- Contrastive loss
- Few-shot learning
- Support set that used for re-training
- Learning new activities on the Edge

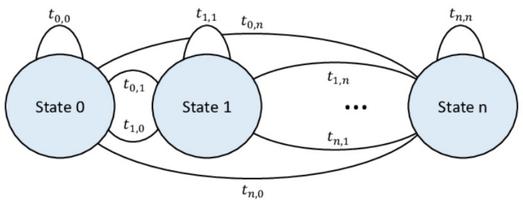
MAGNETO on the Edge











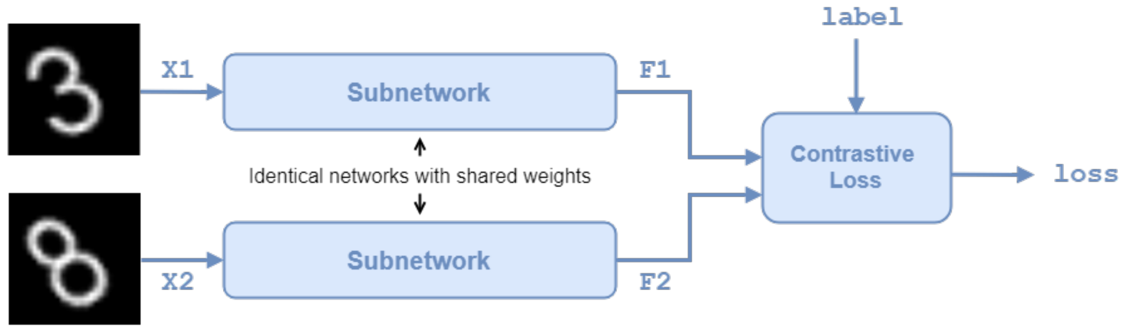
Post processing

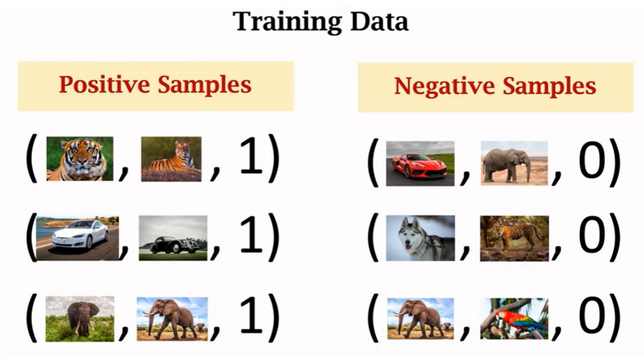
- Majority sequential measurements
- Confidence
- Weighted time window

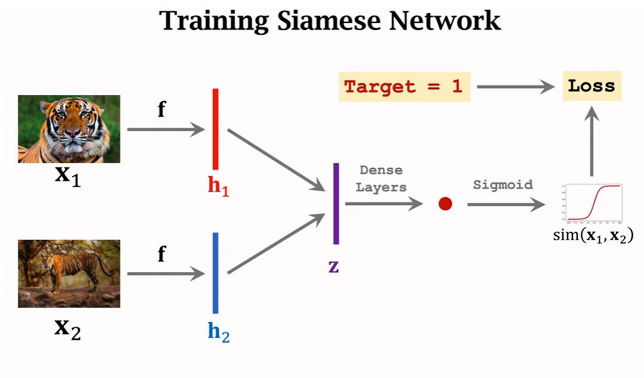
MAGNETO on the Edge

Dynamic extention with new activities

- Train the network to learn similarities
- Different loss functions can be used:
  - *Contrastive*: helps when classes are not all known at training time
  - *Triplet*: triplet of data instead of pairs: Anchor, positive example, negative example







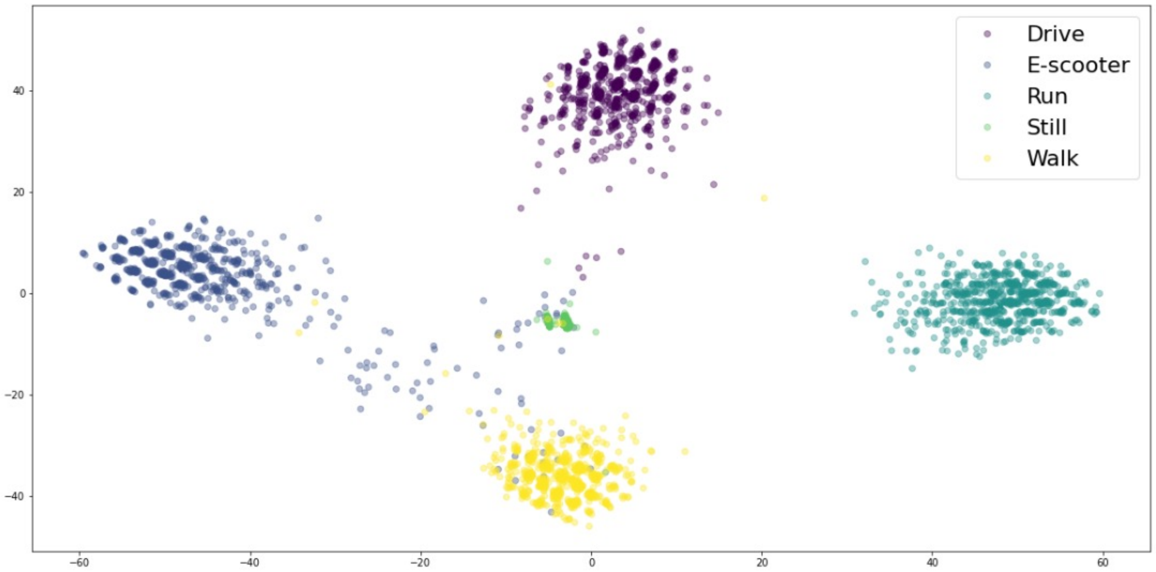
$$L(x_1, x_2, Y) = Y * ||F(x_1) - F(x_2)||^2 + (1 - Y) * \left\{\max\left(0, m^2 - ||F(x_1) - F(x_2)||^2\right)\right\}$$

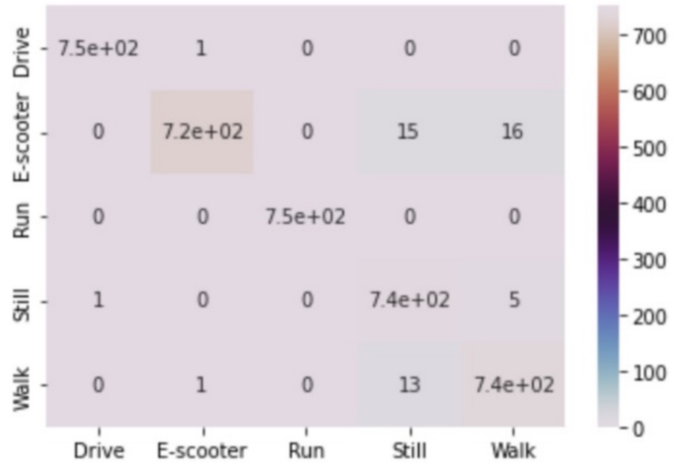
- $Y = \{1$: similar samples , 0 : unsimilar samples $\}$
- m = max distance in the embedding space, that contributes to the loss (we set it to 60 in the experiments)

MAGNETO on the Edge

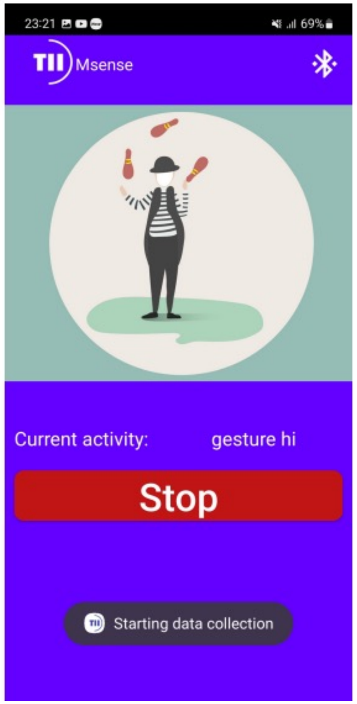
- Network Architecture: *input* (159)×512×124×64×34 (embedding space)
- Train and test using **all** classes
- Accuracy 98.6%

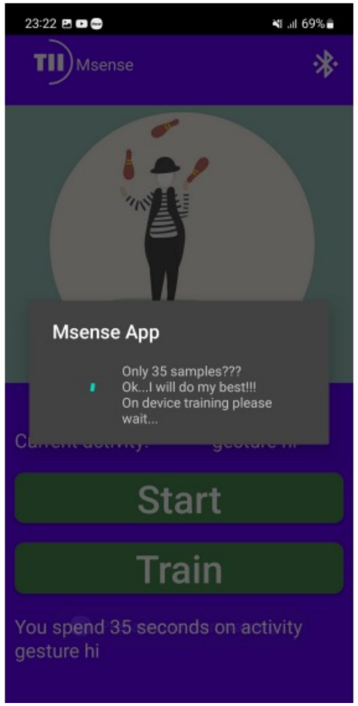
	precision	recall	f1-score	support
0	0.99867	0.99867	0.99867	750
1	0.99723	0.95867	0.97757	750
2	1.00000	1.00000	1.00000	750
3	0.96373	0.99200	0.97766	750
4	0.97226	0.98133	0.97678	750
accuracy			0.98613	3750
macro avg	0.98638	0.98613	0.98613	3750
weighted avg	0.98638	0.98613	0.98613	3750

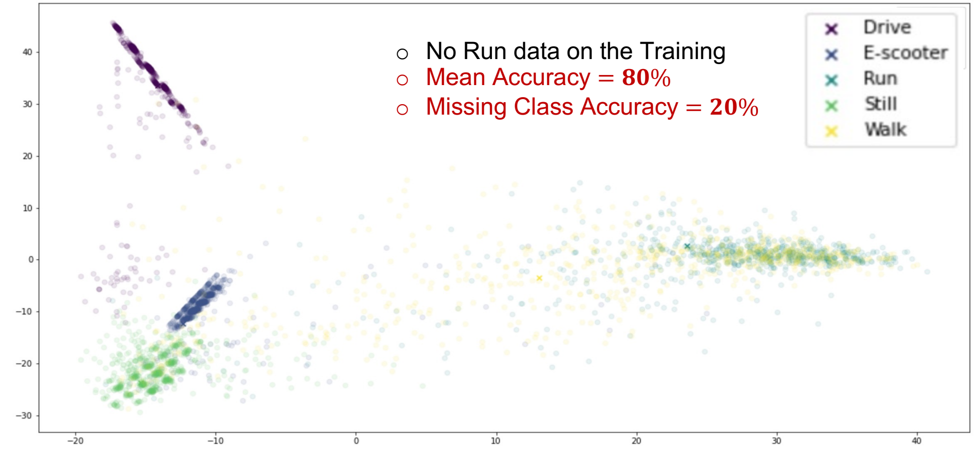


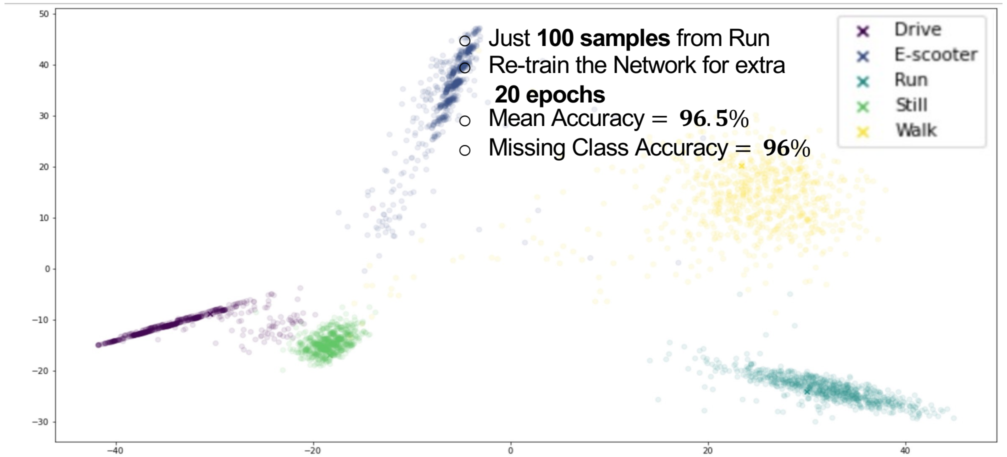


Learning new Activities on the Edge

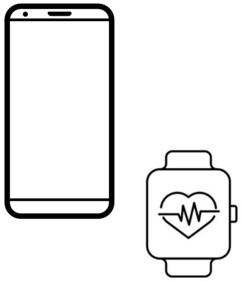


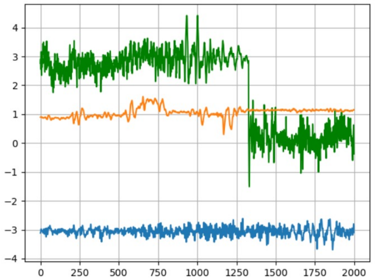


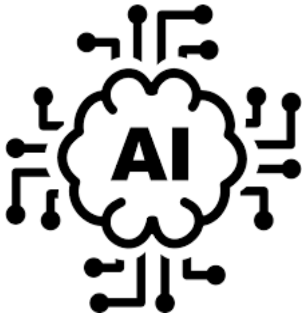


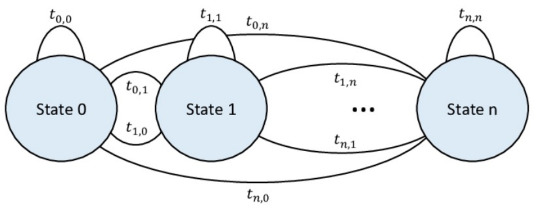


Learning new Activities on the Edge







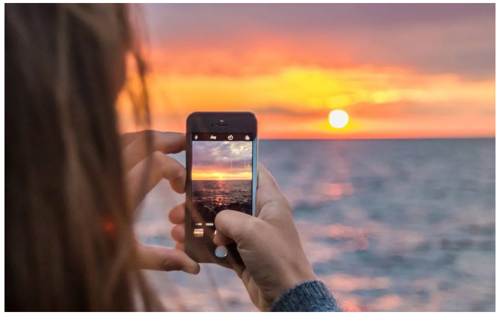


Include: model, support set, newly captured data, feature extraction function, etc...

< 5MB

=

2x



Video Demo comes Next

MAGNETO in a nutshell

Conclusion

- ✓ **Inference** ⇒ On Edge
- ✓ **Learning** ⇒ On Edge
- ✓ **Latency** ⇒ In milliseconds
- ✓ **Users data** ⇒ Private

Future Work

- Catastrophic forgetting on the Edge
- New activities learning needs
- Data augmentation
- Bring it to the end user

Learning on Edge
Challenges and future directions

- Learning Generalization and Adaptation
- Theoretical Foundation
- Architectures for Heterogeneity and Scalability
- Fortified Privacy
- Hybrid Approach
- Data Quality Assurance
- Framework Extension
- Standardization


Thank you!