Confirm Smart Manufacturing

Toward Distributed, Global, Deep Learning Using IoT Devices

Bharath Sudharsan, Pankesh Patel, John Breslin, Muhammad Intizar Ali, Karan Mitra, Schahram Dustdar, Omer Rana, Prem Prakash Jayaraman, Rajiv Ranjan

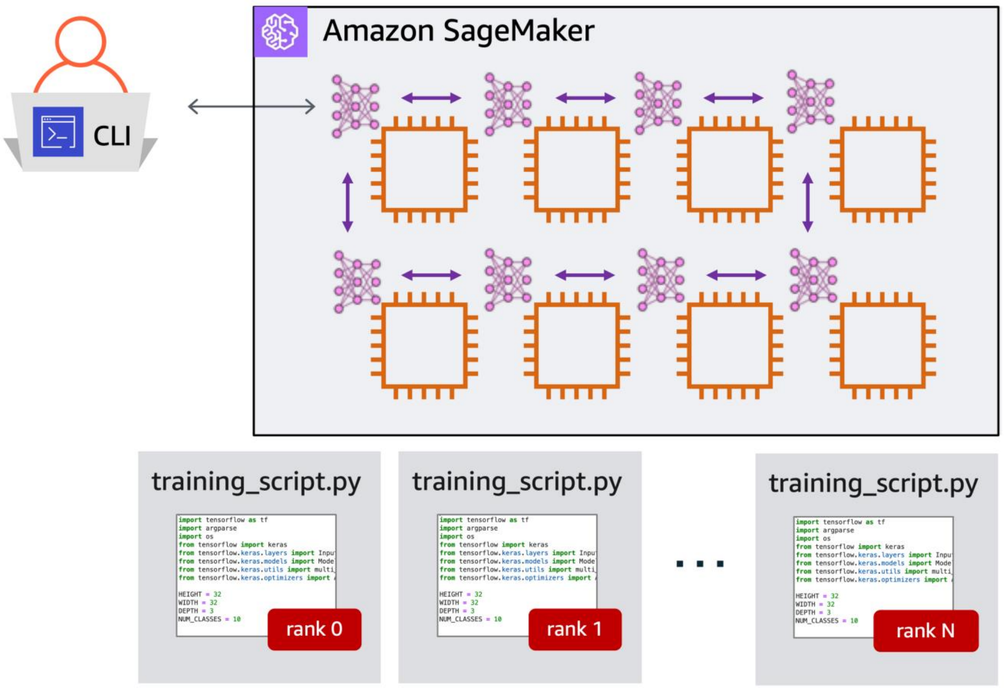
A World Leading SFI Research Centre

Science Foundation Ireland For what's next

University of Limerick · Tyndall National Institute · UCC University College Cork, Ireland · NUI Maynooth · OÉ Gaillimh NUI Galway · Cork Institute of Technology · AIT · LIT

Distributed Training

Confirm
Smart Manufacturing



Distribute training on multiple GPUs using Amazon SageMaker

- In Deep Learning (DL), more is better
  - ➢ More data, layers, compute power, leads to higher accuracy, and better robustness of trained models
- Distributed training can improve model convergence speed
  - ➢ Every GPU runs exact same copy of the training script. Each training process is uniquely identified by its rank
  - ➢ As the number of training processes increases, inter-process communications increases, and communication overhead starts affecting scaling efficiency

IoT Devices - Hardware View

Confirm Smart Manufacturing

ARM Cortex-M0 MCU based BLE beacon

Powerful CPU + basic GPU based SBCs (single board computers)

Edge gateway with GPUs and SSDs

Edge computing hardware: highly resource constrained -> high resource (left to right)

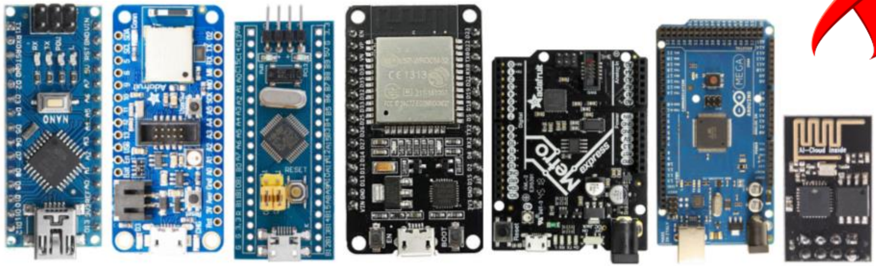
- MCUs and small CPUs: BLE beacons, smart bulbs, smart plugs, TV remotes, fitness bands
- SBCs: Raspberry Pis, BeagleBones, NVIDIA Jetsons, Latte Pandas, Intel NUCs, Google Coral
- GPU accelerated: AWS snowball, Digi gateways, Dell Edge Gateways for IoT, HPE Edgeline

- Roughly **50 billion** MCU chips were shipped in 2020 (market estimates), which far exceeds other chips like GPUs & CPUs (only 100 million units sold)

IoT Devices - Hardware View

Powerful CPU + basic GPU based SBCs (Single Board Computers)

Billions of IoT devices are designed using such MCUs and small CPUs



MCU1	MCU2	MCU3	MCU4	MCU5	MCU6	MCU7
ATmega328P	nRF52840	STM32f103c8	ESP32	ATSAMD21G18	ATmega2560	ESP8266
8 kB SRAM	256 kB SRAM	20 kB SRAM	520 kB SRAM	32 kB SRAM	8 kB SRAM	32 kB SRAM
32 kB Flash	1 MB Flash	128 kB Flash	4 MB Flash	256 kB Flash	256 kB Flash	1 MB Flash
@ 16 MHz	@ 64 MHz	@ 72 MHz	@ 240 MHz	@ 48 MHz	@ 16 MHz	@ 80 MHz

- SBCs for DL model training – established area
  - ✓ Numerous papers, libraries, algorithms, tools exists to enable ML self-learning and re-training
  - ✓ ML framework support i.e., TF Lite can run on SBCs and not on MCUs
- MCUs for DL model training – emerging area
  - ✓ Edge2Train: Train SVMs on MCUs
  - ✓ Train++: Ultra fast incremental learning
  - ✓ ML-MCU: Train 50 class classifier on 3$ IoT chip

Distributed Training on IoT Devices

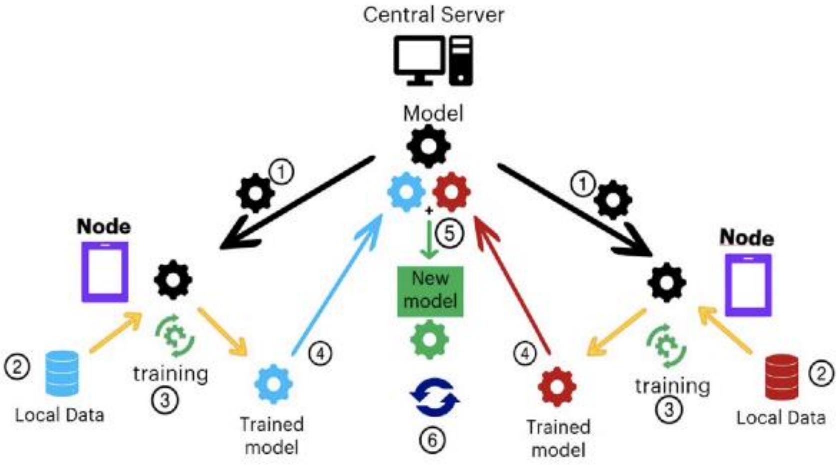
Confirm
Smart Manufacturing

=

2500$ GEFORCE RTX 2080 Ti GPU setup with 11 GB RAM (left). Common Alexa smart speaker with approx. 2 GB RAM each

- When idle IoT devices are efficiently connected, it can collectively train mid-sized models
  - ✓ Efficiently connecting 20 Alexas can collectively pool 40 GB of RAM
  - ✓ Possible to train in similar speeds as GPU, but at a 0 $ investment since millions of IoT devices already exist globally, and most of them are idle

Distributed Training on IoT Devices



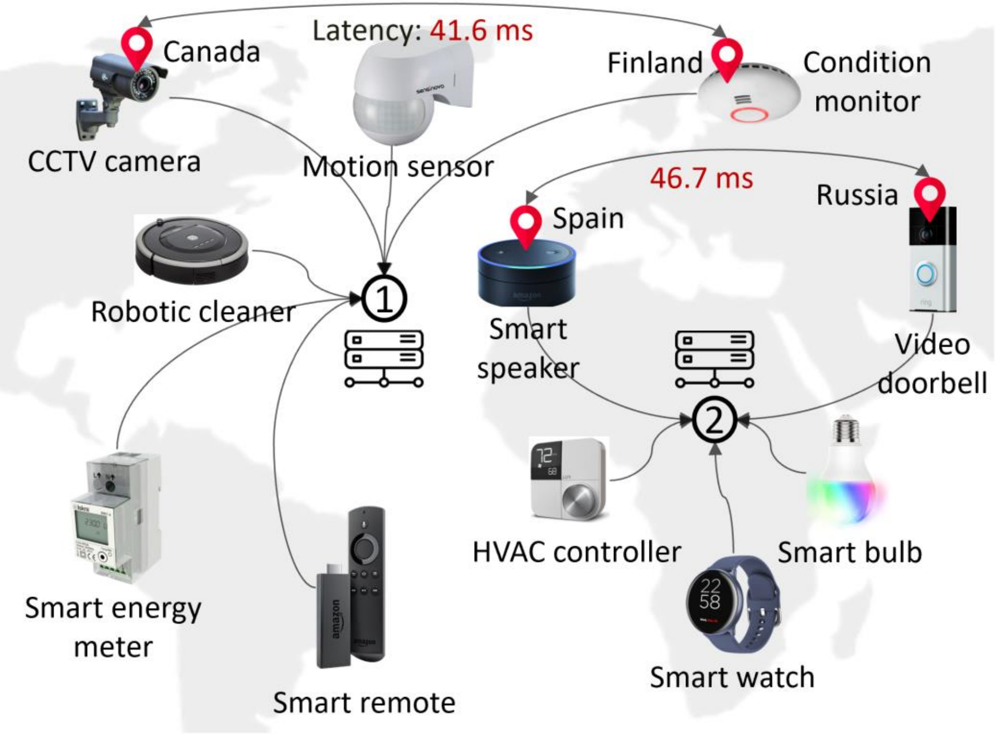
Distributed learning on IoT devices

- With strict privacy regulations, the historic datasets building process is rapidly transforming into *historic intelligence building* - achieved by distributed learning on the IoT devices, then central model combining

- ML model aggregation rather than data aggregation - using locally generated data to collectively train a model without storing or transmitting data to server

- Use case and model combining methods: https://github.com/bharathsudharsan/ML-Model-Combining

Distributed, Global Training on IoT Devices



Servers coordinating with geographically separated IoT devices to produce a trained model

- Distributed training of one DL model on the hardware of thousands of IoT devices is the ***future of ML and IoT***
  - Improved convergence speed
  - Improved data privacy
  - Effective utilization of idle devices
  - Avoid investing on GPU clusters or Cloud
- Global training scenarios/setups can be impacted by real-world network uncertainties and staleness. Challenges are presented in upcoming slides

High FLOPs Consumption

Confirm Smart Manufacturing

- FLOPS = Floating point operations per second. FLOPs = Floating point operations
  - FLOPS is a unit of speed. FLOPs is a unit of amount
- The input/activation of video analytics DL networks has [N, T, C, H, W] as its five dimensions
  - N is batch size, T is temporal timestamps, C is channel number, H & W are spatial resolution
- **Computational overhead and network congestion:** Can apply 2-D CNN to each video image frames - temporal relationship between the frames cannot be modeled/learned
- **More parameters can cause stalling**: For distributed learning of spatio-temporal data, the models with 3-D convolutions, in addition to large model size, also suffers from large number of parameters,
  - Main reason to slow down the training and communication process even within a GPU cluster
  - Training will stall when unexpected network issues are encountered

Slow Exchange of Model Gradients

Confirm Smart Manufacturing

Network conditions across different continents

- Shanghai to Boston: Even at speed of light, direct air distance - still takes 78 ms to send and receive a packet
  - ➢ 11, 725 km × 2/(3 × 10^8 m/s) = 78.16ms. Information collected from Google Maps
- Network conditions across different continents. Different from training inside a data center, long-distance distributed training suffers from high latency, which proposes a severe challenge to scale across the world

Staleness Effects

Confirm Smart Manufacturing

- Network communication bottleneck produce stale parameters
  - Model parameters arrive late, not reflecting the latest updates
  - Staleness during training can lead to model instability
  - Staleness not only slows down convergence but also degrades model performance
  - Popular distributed model training techniques (e.g., SSGD, ASGD, D2, AD-PSGD) adopt a nonsynchronous execution approach to handle staleness
  - Not feasible to monitor and control staleness in the current complex IoT environments containing heterogeneous devices using different network protocols
- Staleness challenges can be addressed by designing *accuracy guaranteeing dynamic error compensation* and network coding techniques

Dataset I/O Efficiency

- Datasets are usually stored in a high-performance storage system (HPSS), shared across all worker nodes
  - ➢ HPSS systems have good sequential I/O performance, their random-access performance is inferior, causing bottlenecks for large data traffic
- Research needs to consider novel data approximation, sampling and filtering methods
  - ➢ Develop a method to identify videos that have multiple similar frames (i.e., we say that nearby frames contain similar information), then load and share only nonredundant frames during distributed training
  - ➢ Similarly, for other datasets associated with images and sensor readings, we recommend filtering or downsampling the data without losing information, then distributing it during training

Design Considerations

Confirm Smart Manufacturing

- **Latency and Bandwidth:** Dynamic and depend on the network condition, which we cannot control
- **Scalability:** Essential when connecting many devices. To improve, we need to significantly reduce communication cost (Tc), which is determined by network bandwidth, latency
  - *Tc = latency + (model size/bandwidth)*
  - If we can achieve X times training speedup on Y machines, the overall distributed training scalability (defined as X/Y) increases
- **IoT Hardware Friendliness:** When following SSDS and ASGD, the IoT devices need to use techniques to tolerate extreme network conditions by reducing the data to be transferred
  - Gradient sparsification, temporally sparse updates, gradient quantization/compression
  - Accommodating such techniques add computation strain while consuming the limited memory that is sufficient only for training models and executing the device's routine functionalities

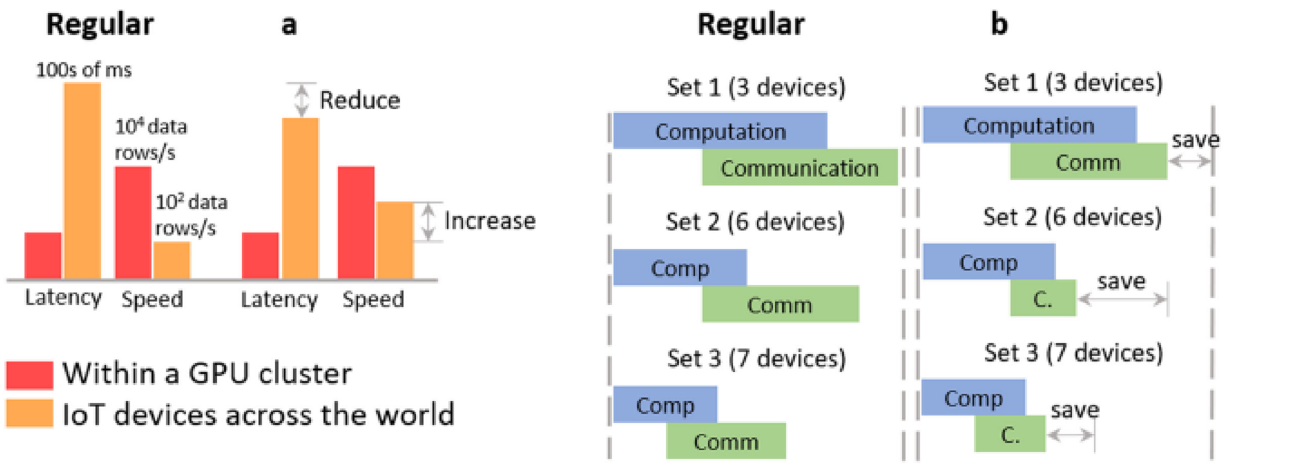
Two-step Deep Compression Method

Confirm Smart Manufacturing

- **Step One:** Sets the weight threshold *Wt* high for the training involved devices that have a poor internet
  - Reduces frequent transmission of weights, reducing the network bandwidth. i.e., weights are locally accumulated till it reaches *Wt*
  - Less usage of congested network by not allowing to transmit weights frequently
- **Step Two:** Shrinks all the accumulated weights using the encode() function
  - Similar to how Vanilla, Nesterov momentum SGD encodes the parameters/gradients
  - Shrink and efficiently transmit trained weights without straining IoT hardware
- Both steps jointly improve *training scalability and speed* by tolerating the *real-world network uncertainties* and by *reducing the communication-to-computation ratio*

Two-step Deep Compression Method

Confirm Smart Manufacturing



Comparing distributed training within a GPU cluster versus training using geographically distributed IoT devices

- The proposed two-step deep compression method can

  (a) Tolerate latency and increase training speed

  (b) Reduce the communication-to-computation ratio to improve scalability and reduce communication costs

Summary

Confirm
Smart Manufacturing

- Presented the concept of training DL models on idle IoT devices, millions of which exist across the world
  - **Reduces Cost**: Avoiding investing in GPU clusters or Cloud by effective utilization of idle IoT devices
  - **Improves Privacy**: Historic datasets building process transforming into historic intelligence building
  - **Improves Training Speed**: Can pool massive hardware resources if devices are efficiently connected
- Identified and studied challenges that impact the distributed, global training on IoT devices
- Presented a two-step deep compression method to improve distributed training speed and scalability
  - Can significantly compress gradients during the training of a wide range of NN architectures
  - Can also be utilized alongside TF-Lite and Federated Learning approaches thereby providing the basis for a broad-spectrum of decentralized and collaborative learning applications

Confirm
Smart Manufacturing

Confirm
Smart Manufacturing

Contact: Bharath Sudharsan
Email: bharath.sudharsan@insight-centre.org

www.confirm.ie

Science Foundation Ireland
SFI
For what's next