Confirm
Smart Manufacturing

Globe2Train: A Framework for Distributed ML Model Training using IoT Devices Across the Globe

Bharath Sudharsan, John G. Breslin, Muhammad Intizar Ali

A World Leading SFI Research Centre

Science Foundation Ireland For what's next

Introduction

Confirm Smart Manufacturing

- Every modern household does not compulsorily own a GPU, yet it roughly has around a dozen IoT devices
  - When idle IoT devices efficiently connected, can locally, within their home network train mid-sized ML models without depending on Cloud or GPU servers
  - Latest GEFORCE RTX 2080 Ti GPU has 11 GB RAM but 1500 $. Whereas one Alexa smart speaker has 2 GB RAM and efficiently connecting 20 can collectively pool 40 GB of RAM. In this way, we can complete training faster than the expensive GPU and at a 0 $ investment since millions of IoT devices already exist globally, and most of them are idle
- Challenges in distributed global training scenarios:
  - **Network Uncertainties: Congestion and Latency Variance** in real-world IoT networks
  - **Staleness Effect:** stale parameters. i.e., the model parameters arrive late, not reflect the latest updates. Staleness slows down convergence

Related Papers

Confirm
Smart Manufacturing

TABLE I
COMPARISON OF G2T WITH THE MOST RELATED PAPERS.

Paper	Experimentation hardware	Training reliability	On-device algorithm	MCUs & small CPUs support	Staleness	Uncertainties toleration	Scalability	Parameters compression
[1]	Distributed AWS resources	Not Investigated (NI), server don't fail	Non Comparable (NC)	✗	Reduced	✓	Few distributed servers	Delayed updates
[2]	Multiple GPUs	NI, GPUs in cluster also don't fail	NC	✗	NC	✗	Hundreds of GPUs in a cluster	✗
[3]	Mobiles	NI	ML framework-based	✗	Reduced	✓	Millions of mobiles	Sparse updates
[4]	Edge servers, Rpis	Not Available (NA)	Cannot run on MCUs	✗	NI	✗	NC	Pruning, Quant
[5]	Laptops, Rpis	NA	Cannot run on MCUs	✗	NI	✗	NC	✗
G2T (ours)	MCUs that are widely used to design IoT devices	G2T-C counter balances for device/network failures	G2T-D is designed for resource-friendly training, transmission even in poor network	All chipsets in Arduino IDE, Atmel studio, ARM Keil MDK	G2T-C is designed not to face staleness	Tolerates un avoidable latency, congestion, bandwidth issues	Thousands of distributed heterogeneous IoT devices	Resource-friendly shrinking by G2T-D to reduce bandwidth

[1] L. Zhu, Y. Lu, Y. Lin, and S. Han, "Distributed training across the world," 2019

[2] J. Lin, C. Gan, and S. Han, "Training kinetics in 15 minutes: Large-scale distributed training on videos,"arXiv preprint, 2019

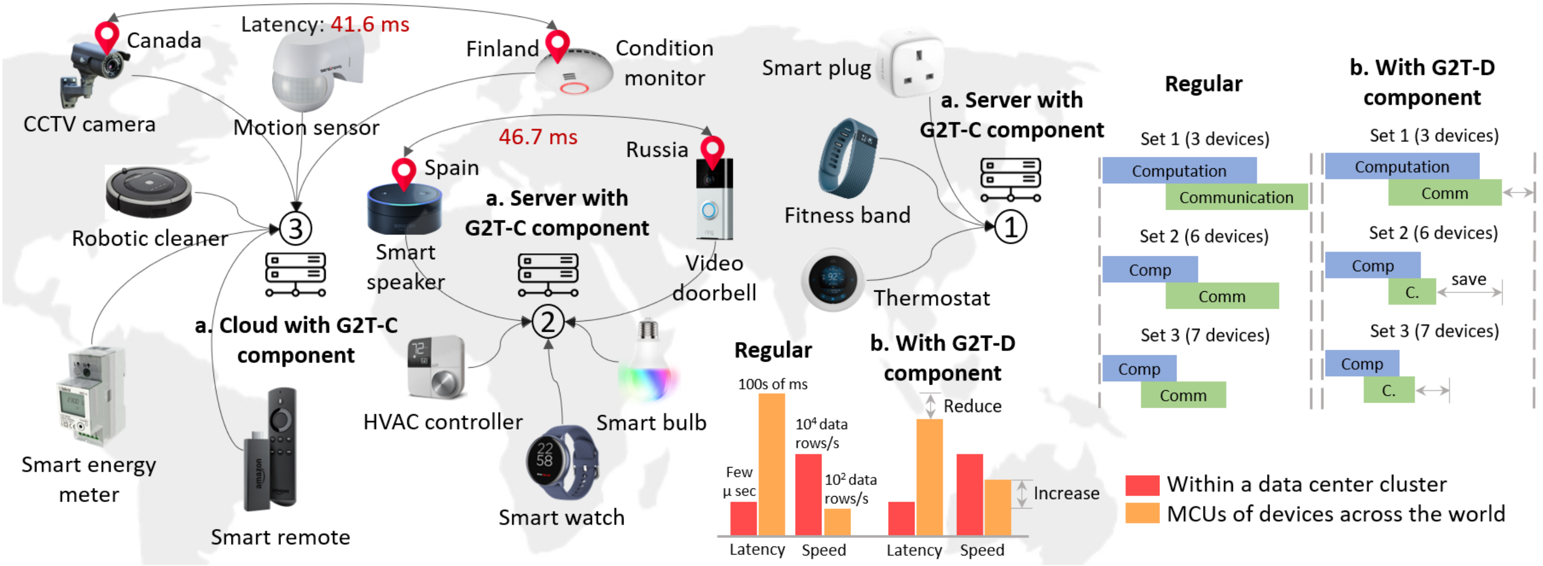
[3] Y. Lin, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training,"arXiv preprint, 2017

[4] W. Xu, and N. Xiong, "Accelerating federated learning for iot in big data analytics with pruning, quantization and selective updating, "IEEE Access, 2021

[5] S. Wang, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in IEEE INFOCOM, 2018

Globe2Train

Confirm Smart Manufacturing

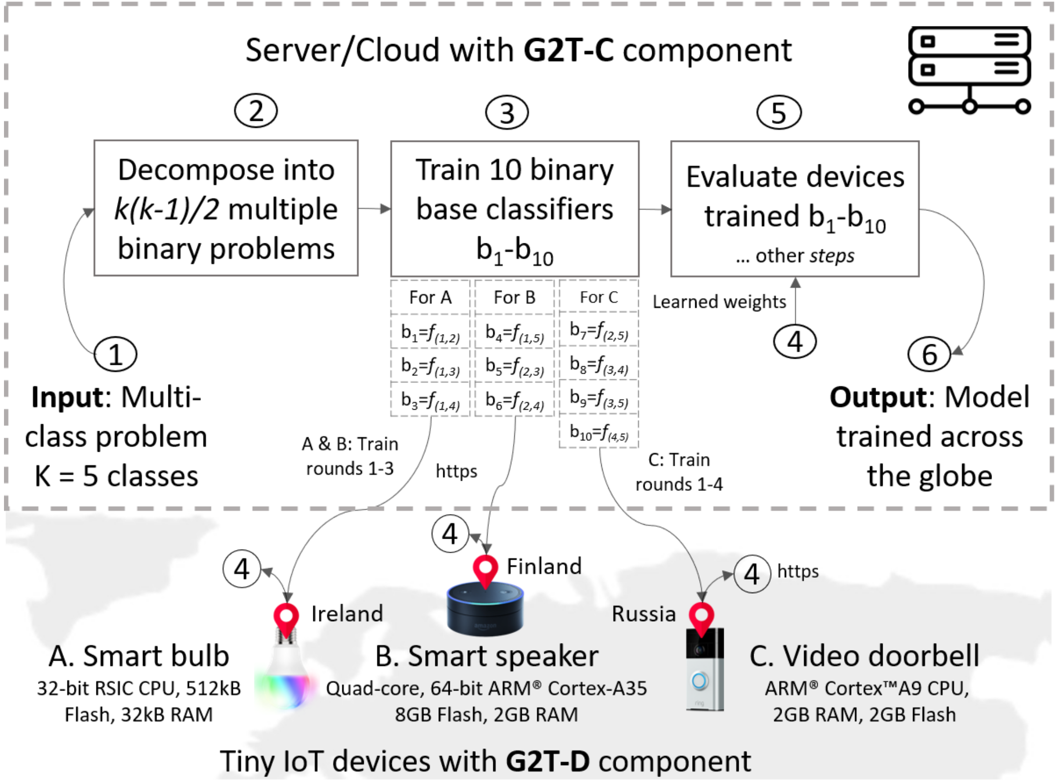


- The Globe2Train (G2T) framework enables distributed ML model training using IoT devices across the globe:

  a. The framework's G2T-Cloud (G2T-C) component is deployed on a central server/cloud

  b. The G2T-Device (G2T-D) component is deployed on training involved IoT devices

G2T-C Component



- G2T-C component decomposes one multi-class problem into multiple binary problems, which IoT devices solve and updates back the weights
  - Decomposes 1 multi-class problem into k(k-1)/2 binary problems
  - Smart speaker trains 3 binary classifiers. b_4 (to differentiate class 1 data from class 5 data), b_5 (class 2 from 3) and b_6 (2 from 4)

G2T-D Component

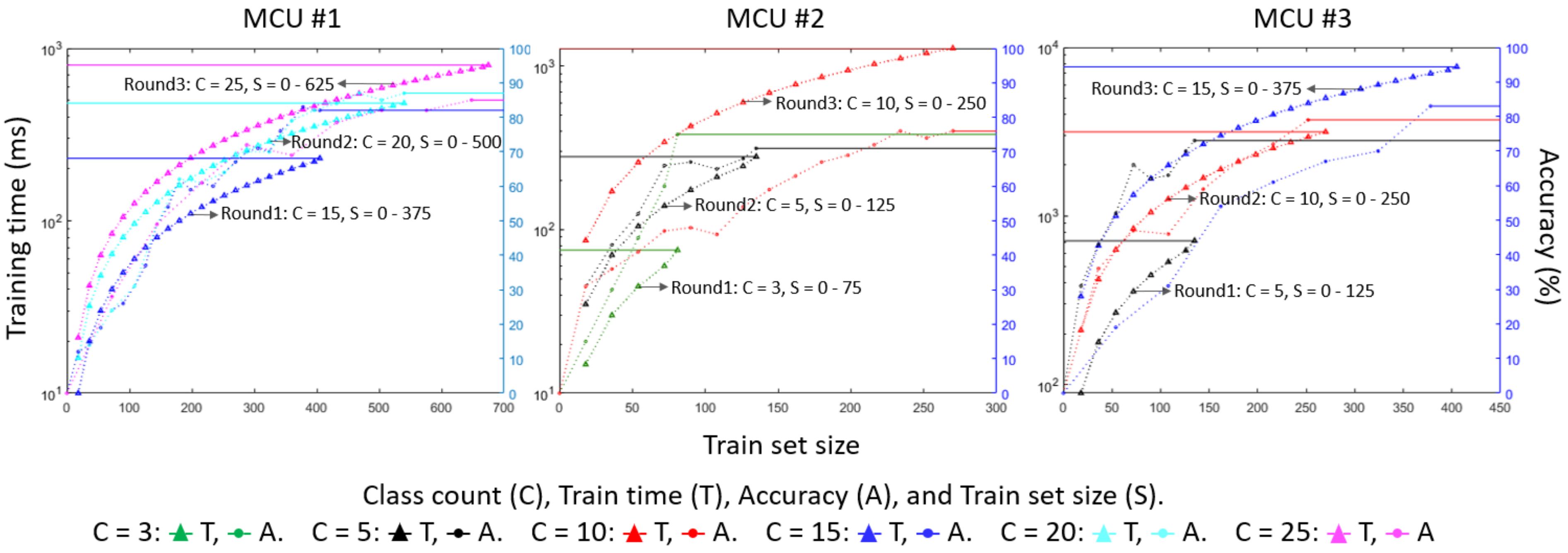
Algorithm 1 G2T-D to efficiently transmit the IoT device trained weights even under poor network conditions.

1: **Input**: Available weights $w_0, \ldots, w_n$ of IoT device trained base binary classifiers $b_1, \ldots, b_n$
2: **Output**: Efficiently transmit shrunken weights
3: $w_{s0}, \ldots, w_{sn}$ to a central server with G2T-C component
4: Set weights threshold $W_t \leftarrow 1/ \propto$ network quality
5: **if** n in $w_n > W_t$
6:     **for** i = 0, 1, 2, …, n **do**
7:         $w_{si} \leftarrow$ shrink the weight w_i. Use *encode* (w_i)
8:     **end for**
9: Transmit $w_{s0}, \ldots, w_{sn}$. Use HTTPS
10: **else**
11:     Locally accumulate weights till W_t

- line 4: Sets the weight threshold W_t high for the training involved devices that have a poor internet connection
  - ➢ Reduces frequent transmission of weights, reducing the network bandwidth. i.e., weights are locally accumulated till it reaches W_t
- Lines 6 to 8: Shrinks all the accumulated weights using the encode() function, which packs the non-zero weight values
  - ➢ Similar to how Vanilla, Nesterov momentum SGD encodes the parameters/gradients

Results

Confirm
Smart Manufacturing



- We initiate distributed training from G2T-C component, where it decomposes the given 95 classes problem into 4465 binary problems (i.e., K = 95 in K(K-1)/2), where b_0 to b_{4465} binary classifiers need to be trained by MCUs 1-3 in numerous rounds and in each round report back the calculated weights

Results

Confirm
Smart Manufacturing

- Contributions of the G2T-D Framework Component
  - Congestion and Latency Toleration: Reduces weights synchronization frequency (less usage of congested network) by not allowing to transmit weights frequently or based on its availability. So, the training process used in across the globe setting ***gains the ability to tolerate latency***
  - Improved Training Scalability: Sends the weights at intervals that depend on the network condition - this process ***improves scalability by reducing the communication-to-computation ratio***
  - IoT Hardware Friendliness: Implementation is only a few lines of code
- Contributions of the G2T-C Framework Component
  - Overcoming Staleness Effect: Designed not to face staleness and is more efficient than SSGD and ASGD - as it intelligently splits the tasks that execute in multiple rounds on the IoT devices

Conclusion

Confirm Smart Manufacturing

- We presented Globe2Train, a framework for training ML models on idle IoT devices, millions of which exist across the globe
- G2T-C and G2T-D framework components can improve distributed training scalability, speed while reducing communication frequency and tolerating network latency
- Since the G2T-C can decompose one multi-class problem into multiple binary problems, it can be used to decompose a resource-demanding problem (can run only on GPU clusters) into multiple resource-friendly tiny parts that can distributedly execute on numerous idle IoT devices across the globe
- Since the G2T-D can significantly compress gradients/parameters during distributed training, it can be the basis for a broad spectrum of decentralized and collaborative learning applications

Future Work

Confirm
Smart Manufacturing

- Perform comprehensive real-world experimental evaluation of G2T
  - Deploy the G2T-D on a few geographically separated IoT devices - poor to good network conditions
  - Deploy the G2T-C on an Amazon AWS server and establish communication with IoT devices
  - Define a ML model on AWS, then instruct the G2T-C to decompose the given ML multi-class problem into multiple binary problems and assign it to the connected IoT devices
  - Make IoT devices solve the binary problems (by performing model training), then using G2T-D report back the calculated weights
  - Now, we should have a full ML model, that was distributedly trained by multiple IoT devices and capable to solve a multi-class problem

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