The 4th Italian Conference on Big Data and Data Science (ITADATA 2025)
09-11 September 2025 – Turin, Italy

BIG DATA & COMPUTE CONTINUUM RESEARCH LAB

Investigating Edge Fine-Tuning of Large Language Models in a Federated Environment

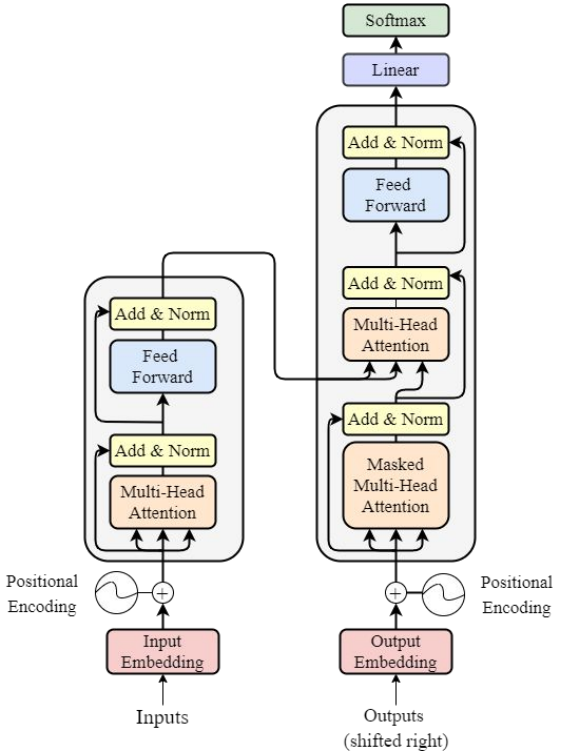
Lorenzo Colombi, Michela Vespa, Francesco Resca, Sara Cavicchi, Edoardo Di Caro,
Elena Bellodi, Mauro Tortonesi and Cesare Stefanelli

University of Ferrara, Ferrara, Italy

Università degli Studi di Ferrara

Background: LLM

- **Large Language Models** (LLM) are deep **neural networks** trained on **massive amounts** of textual **data** to understand and generate **human-like language**.
- The core **architecture** behind almost all modern LLMs is the **Transformer** which leverages the **attention** mechanism
- LLMs could be **fine-tuned** on **specific tasks** through further training on a targeted dataset.

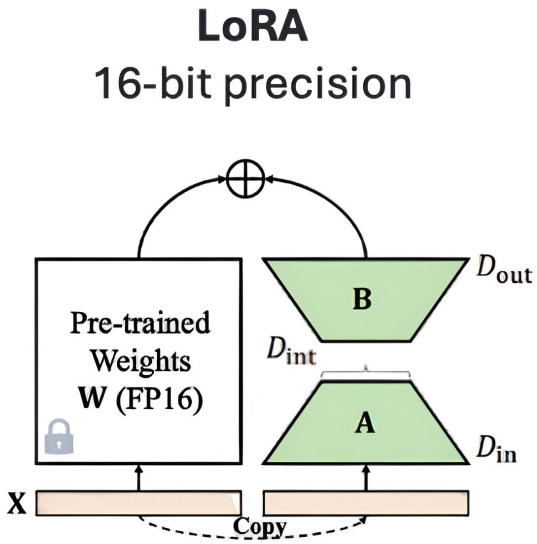


Background: Parameter-Efficient Fine Tuning (PEFT)

- **PEFT** allows a **resource-efficient fine-tuning**, only updating a small subset of model's parameters.

- **Advantages**:
  - Reduces computational costs
  - Speeds up the training process
  - Lowers memory consumption
    (GPT-3 (175B): 1TB -> 25MB checkpoint)

- **Low-Rank Adaptation (LoRA)** freezes the original weights and introduces adapters, which are trained and then fused with the original parameters.



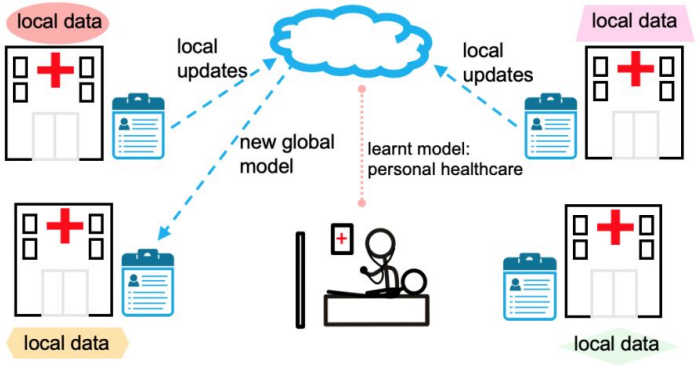
Background: Small Language Models (SLMs)

- **SLMs** are language models with fewer parameters, usually less than 10B.
- They are derived from larger model using **compression** techniques such as **distillation** or **quantization.**
- **Lightweight** and **efficient**, SMLs can run on **edge** or resource-constrained devices while maintaining acceptable performance.

Model	MMLU	TriviaQA	ARC	PIQA	Hellaswag	OBQA	Winogrande	Avg
number of parameters < 200M								
GPT2-137M	26.29	0.49	31.09	62.51	29.76	29.40	49.72	32.75
Pythia-160M	26.68	0.34	31.92	61.64	29.55	27.80	49.49	32.49
MobileLM-125M	-	-	35.51	65.30	38.90	**39.50**	**53.10**	-
SmolLM-135M	**30.23**	**4.11**	**43.99**	**69.60**	**42.30**	33.60	52.70	39.50
number of parameters < 500M								
GPT2-medium-380M	27.77	2.37	34.30	66.38	37.06	31.20	49.49	35.51
Qwen1.5-500M	30.34	0.21	40.92	69.31	48.22	33.60	52.64	39.32
Qwen2-500M	31.92	1.47	39.74	69.31	47.61	33.00	54.14	39.60
MobileLM-350M	-	-	43.65	68.60	49.60	**40.00**	**57.60**	-
SmolLM-360M	**34.17**	**9.10**	**51.10**	**72.00**	**53.80**	37.20	53.70	44.44
number of parameters < 2B								
Cosmo-1B	32.39	4.75	44.05	71.38	52.70	35.40	54.54	42.46
Olmo-1B-hf	32.15	4.71	44.17	75.14	60.49	38.40	53.28	44.05
GPT2-xl-1.5B	30.10	6.00	39.78	70.51	47.91	34.40	53.12	40.26
Karpathy/GPT2-1.5B	38.06	13.94	56.60	75.79	60.43	41.80	55.80	48.92
Qwen1.5-1.8B	33.46	0.95	47.08	74.32	59.55	37.00	58.41	44.40
Qwen2-1.5B	37.87	13.70	48.20	75.46	63.86	36.80	59.98	47.98
MobileLLM-1.5B	-	-	54.20	74.80	**64.50**	**50.05**	64.70	-
Phi-1.5	35.90	6.05	60.59	75.52	60.72	46.00	**67.72**	50.36
SmolLM-1.7B	**39.97**	**22.25**	**61.55**	**77.30**	64.10	42.60	56.00	**51.97**

Background: Federated Learning (FL)

- It could be use in scenarios where **centralize the data is not possible**.
- FL **clients** locally **train their** own **model**. The **server** collects the parameters from the clients and **aggregates** them updating the **global model**.
- **Federated Averaging** aggregates the weight computing a **weighted average** based on the **number of samples** used by each client.



Background: Federated Averaging (FedAvg)

- Despite being the most commonly used weight aggregation technique **FedAvg** presents several **shortcomings**.
- The weight **average** is only based on the **number of samples** used by the client, without taking into account the distribution and the nature of the sample.
- Especially **when training LLM** the **quality** and **lengths** of the samples could have a **impact** on the overall performance

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**

$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$

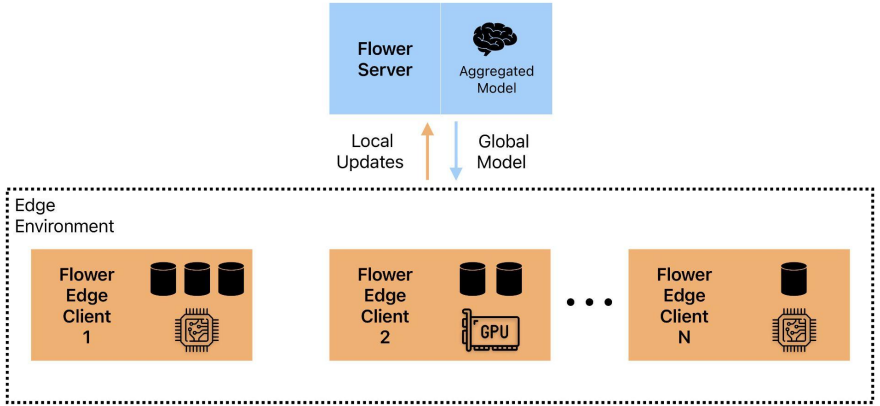
$m_t \leftarrow \sum_{k \in S_t} n_k$

$w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{m_t} w_{t+1}^k$ *// Erratum*[4]

$$\min_w F(w), \text{ where } F(w) := \sum_{k=1}^{m} p_k F_k(w)$$

Token-based Federated Averaging

- Token-based Federated Averaging is a weighting scheme that adjusts each client's contribution using a **linear combination** of the **number of tokens** and **training examples** processed **locally**.
- Approach particularly valuable in federated settings where **sample length** is **highly heterogeneous**.
- Varying the value of **α** modifies how the **number of tokens** and the number of **samples influence** the overall weights.



$$\mathbf{w}_{\tau+1} = \mathbf{w}_\tau - \eta \sum_{k=1}^{K} ((1-\alpha)\frac{n_k}{N} + \alpha\frac{t_k}{T})\, \mathbf{w}_t^{(k)}$$

Experimental Evaluation

- As LLMs we used the **135M** and **360M** versions of the **SMOLLM2** model and we evaluated them using **Bert** and **Rouge score**.
- We compared the performance of FedAvg with the token-based aggregation approach.
- We collected the samples from different open-source dataset.

Client	Dataset	#Examples	Batch size
A	pubmed_qa_211k	32,768	4
B	medical_meadow_medical_flashcards_34k	4,096	1
C	medical_meadow_wikidoc_patient_information_6k + medical_meadow_wikidoc_10k + chatdoctor_icliniq_7k	4,096	1
D	long examples	16,384	4
E	short examples (1)	8,192	1
F	short examples (2)	8,192	1

Experimental Evaluation: first experiment

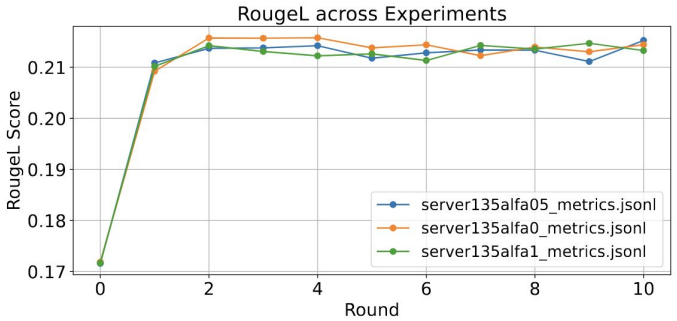
First, we assigned to **each client** a subset of examples collected from a **different dataset** to simulate a distributed scenario where each client can only use the data collected locally.

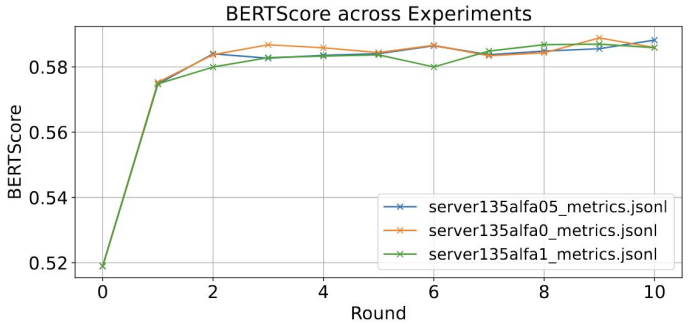
Specifically, client config **A**, **B** and **C** has been used.

Client	Dataset	#Examples	Batch size
A	pubmed_qa_211k	32,768	4
B	medical_meadow_medical_flashcards_34k	4,096	1
C	medical_meadow_wikidoc_patient_information_6k + medical_meadow_wikidoc_10k + chatdoctor_icliniq_7k	4,096	1
D	long examples	16,384	4
E	short examples (1)	8,192	1
F	short examples (2)	8,192	1

Experimental Evaluation: first experiment - results

SmolLM2-360M								
Round	Alpha = 0		Alpha = 0.5		Alpha = 1		Centralized	
	RougeL	BERTScore	RougeL	BERTScore	RougeL	BERTScore	RougeL	BERTScore
0	0.176	0.533	0.176	0.533	0.176	0.533	0.176	0.533
1	0.214	0.590	0.214	0.589	0.214	0.588	0.204	0.579
2	0.216	0.595	0.214	0.594	0.215	0.592	0.206	0.586
3	0.218	0.598	0.216	0.595	0.217	0.593	0.205	0.587
4	0.218	0.597	0.150	0.596	0.217	0.595	0.205	0.588
5	0.218	0.599	0.218	0.598	0.220	0.597	0.207	0.590





Experimental Evaluation: second experiment

The goal was to better highlight the global (aggregated) model **difference** when it the average is **weighted** on the number of **examples** and the number of **tokens**.

Therefore, we merged all the datasets in a single one and we **separated** the **examples** based on the **number of tokens**.

Specifically, client config **D**, **E** and **F** has been used.

Client	Dataset	#Examples	Batch size
A	pubmed_qa_211k	32,768	4
B	medical_meadow_medical_flashcards_34k	4,096	1
C	medical_meadow_wikidoc_patient_information_6k + medical_meadow_wikidoc_10k + chatdoctor_icliniq_7k	4,096	1
D	long examples	16,384	4
E	short examples (1)	8,192	1
F	short examples (2)	8,192	1

Experimental Evaluation: second experiment - results

SmolLM2-135M				
Round	Alpha = 0		Alpha = 0.25	
	RougeL	BERTScore	RougeL	BERTScore
0	0.173	0.521	0.173	0.521
1	0.210	0.575	0.211	0.574
2	0.214	0.584	0.213	0.593
3	0.219	0.591	0.217	0.585
4	0.215	0.591	0.216	0.591
5	0.217	0.591	0.220	0.590

SmolLM2-360M								
Round	Alpha = 0		Alpha = 0.1		Alpha = 0.25		Centralized	
	RougeL	BERTScore	RougeL	BERTScore	RougeL	BERTScore	RougeL	BERTScore
0	0.180	0.533	0.180	0.533	0.180	0.533	0.180	0.533
1	0.221	0.594	0.218	0.596	0.216	0.593	0.207	0.578
2	0.221	0.599	0.217	0.596	0.217	0.595	0.211	0.587
3	0.223	0.601	0.216	0.596	0.217	0.597	0.210	0.588
4	0.224	0.603	0.219	0.601	0.219	0.599	0.210	0.590
5	0.225	0.604	0.219	0.601	0.221	0.600	0.213	0.590

Conclusion and Future Work

Our proposed **token-based aggregation** for LLMs on edge devices in **federated** settings achieved **competitive** (sometimes superior) **performance** vs centralized & FedAvg. However, **improvements** remain **marginal**.

This could be possibly explain with the extremely **limited number of parameters** of the selected LLMs and the **lack of diversity**, in term of number of tokens, in the used dataset.

Future work includes extending to broader datasets, especially with **dataset** with **more diverse** example **length**, use cases, and explore **LLMs** as semantic **evaluators** to overcome the limitations of Rouge and BertScore.

Lastly, we plan to assess the **scalability** of the approach (training time, communication overhead).

Thank you for your attention!

Lorenzo Colombi

PhD student @ **University of Ferrara**

https://ds.unife.it/team/lorenzo-colombi

lorenzo.colombi@unife.it