Challenges in social-aware decentralized AI

Chiara Boldrini*, Luigi Palmieri, Andrea Passarella and Lorenzo Valerio

IIT-CNR, Via G. Moruzzi 1, Pisa, 56124, Italy

Abstract

The recent wave of machine-learning (ML) based Artificial-Intelligence (AI) technologies is having a huge societal and economic impact, with AI being (often silently) embedded in most of our everyday experiences. The research community (and society in general) has already realized that the current centralized approach to AI, whereby our personal data are centrally collected and processed through opaque ML systems, is not an acceptable and sustainable model in the long run. In this work, we illustrate the benefits and challenges of fully decentralized learning by focusing on the specific scenario of "model gossiping" for accomplishing a decentralized learning task, and we study how well global models emerging from the combination of local models perform, where the combination takes into account the social relationships between the nodes (humans associated with the AI). We exploit a custom-built Python simulator that takes as input a social graph and merges local AI models according to a socially-weighted FedAvg-style approach. We show that strategies that work well for federated learning suffer in a fully decentralized environment. Finally, we discuss the current research directions pursued by our group, aimed at addressing the main challenges of decentralized AI.

Keywords

fully decentralized learning, social awareness, data privacy

1. Introduction

The recent wave of machine-learning (ML) based Artificial-Intelligence (AI) technologies is having a huge societal and economic impact, with AI being (often silently) embedded in most of our everyday experiences (such as virtual assistants, tracking devices, social media, recommender systems). The research community (and society in general) has already realized that the current centralized approach to AI, whereby our personal data are centrally collected and processed through opaque ML systems ("black-boxes"), is not an acceptable and sustainable model in the long run. In the CHIST-ERA SAI project[1], we posit that the "next wave" of ML-driven AI should be (i) human-centric, (ii) explainable, and (iii) more distributed and decentralized (i.e., not centrally controlled). These principles address the societal and ethical expectations for trustworthy, privacy-respectful AI, such as those recommended at the European level. They also fit a clear trend to develop decentralized ML for strictly technical reasons, e.g., performance, scalability, and real-time constraints.

Federated learning [1] has been the first mainstream learning approach that moves in this direction. User data are not anymore transferred to a central server. Instead, the server sends a seed AI model to the devices, the model is trained by the devices on the local data, then sent back to the server, which takes care of combining the updates (i.e., the model once trained on local data) received by individual devices into the global model. This is repeated several times until the loss of the model at the central server (which is the result of repeated aggregations of the models locally trained on the user devices) is satisfactorily minimized. However, training on user devices (such as smartphones) brings novel challenges to AI: devices are heterogenous (different capabilities, different operating systems, etc.) and typically have limited resources (in terms of computation, communication, storage, ...). Moreover, the data on the user devices mirror the worldview of the individual user: thus, local datasets are generally non-IID (i.e., classes may be over- or underrepresented in the local data with respect to the general population). This heterogeneity in data distribution can make it challenging to train a robust and accurate machine learning model.

Although federated learning has gained momentum in recent years, a federation of learning devices is still highly dependent on the centralized controller that coordinates the learning efforts. In order to realize the vision described earlier, a *fully decentralized architecture* is needed. Getting rid of the central controller implies that nodes have to coordinate autonomously or that techniques that do not require coordination must be employed. This lack of centralized control can make it difficult to manage and optimize the learning process.

While, from the communication standpoint, one could even assume that all nodes *could* communicate with all other nodes (e.g., via pervasive high-speed 5G technologies), it is not reasonable to assume that all nodes *will*

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy

*Corresponding author.

chiara.boldrini@iit.cnr.it (C. Boldrini); luigi.palmieri@iit.cnr.it (L. Palmieri); a.passarella@iit.cnr.it (A. Passarella); lorenzo.valerio@iit.cnr.it (L. Valerio)

0000-0001-5080-811 (C. Boldrini); 0000-0002-1694-612X (A. Passarella); 0000-0001-5574-7847 (L. Valerio)

[1] https://www.sai-project.eu/

communicate with all other nodes, primarily due to lack of trust, or simply lack of knowledge/opportunities for communications. Therefore, we assume that only certain links between nodes can be activated for communication. The connections between nodes are typically represented with a graph, and nodes can only talk to those that are directly connected to them. Since, in this scenario, no trusted authority is involved, the issue of trust becomes pervasive. In this work, we assume that an edge in the graph represents a shared social relationship between the users, and we take social intimacy as a proxy for trust. The idea is to have nodes exchange knowledge (in the form of AI models) along the edges, combine this knowledge, and reshare it with the graph neighbors. The learning process thus becomes an iterative information diffusion process on a graph. This implies that the graph topology may play a key role in how knowledge flows.

In this work, we illustrate the challenges of fully decentralized learning by focusing on the specific scenario of model "gossiping" for accomplishing a decentralized learning task, and we study what models emerge from the combination of local models, where the combination takes into account the social relationships between the nodes (humans associated with the AI). We show that strategies that work well for federated learning suffer in a fully decentralized learning scenario, by running experiments with a custom-built Python simulator that takes as input a social graph and merges local AI models according to a socially-weighted FedAvg-style approach. The simulator has been developed within the SAI project [2].

2. System model

Let us start with some preliminary definitions. We represent the social network connecting the client devices as $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ the set of edges. We denote with ω_{ij} the trust/social intimacy between nodes i and j. The self-trust ω_{ii} is a pseudo-parameter with which we capture the importance placed by node i on its local model. We assume that only nodes sharing an edge are willing to collaborate with each other: effectively, we use the existence of a social relationship as a proxy of trust.

Each node $i \in \mathcal{V}$ is equipped with a local training dataset $\mathcal{D}_i$ (containing tuples of features and labels $(x, y) \in \mathcal{X} \times \mathcal{Y}$) and a local model h_i defined by weights $\mathbf{w}_i$, such that $h_i(\mathbf{x}; \mathbf{w}_i)$ yields the prediction for input $\mathbf{x}$. Let us denote with $\mathcal{D} = \bigcup_i \mathcal{D}_i$ and with $\mathcal{P}$ the label distribution in $\mathcal{D}$. In general, $\mathcal{P}_i$ (i.e., the label distribution of the local dataset on node i) will be different from $\mathcal{P}$. This captures a realistic non-IID data distribution.

At time 0, the model $h(\cdot; \mathbf{w}_i)$ is, as usual, trained on local data, by minimizing a target loss function ℓ – i.e., $\mathbf{w}_i = \text{argmin}_{\mathbf{w}} \sum_{k=1}^{|\mathcal{D}_i|} \ell(y_k, \mathbf{w}\mathbf{x}_k)$, with $(y_k, \mathbf{x}_k) \in \mathcal{D}_i$. In the following, we refer to $h(\cdot; \mathbf{w}_i)$ as the *isolated local model*. The goal of decentralized learning is to improve the isolated local models h_i by building a local model f_i that takes in information from both the isolated local model h_i and the local models f_j from other nodes. We assume that nodes entertain a certain number of communication rounds, where they exchange and combine local models. Periodically, the device receives the local model from its neighbors in the social graph (hence the *social AI gossip* name, since models are exchanged as a sort of word-of-mouth between nodes). Specifically, at each step t, the local model and the local models from the graph neighbors are averaged. The update function is achieved by averaging the model weights as follows:

$$\mathbf{w}_i(t) \leftarrow \frac{\sum_{j \in \mathcal{N}(i)} \omega_{ij} \alpha_{ij} \mathbf{w}_j(t-1)}{\sum_{j \in \mathcal{N}(i)} \omega_{ij}}, \qquad (1)$$

where we have denoted with $\mathcal{N}(i)$ the neighborhood of node i including itself and α_{ij} is equal to $\frac{|\mathcal{P}_j|}{\sum_{j \in \mathcal{N}_i} |\mathcal{P}_j|}$ (and captures the relative weight of the local dataset of node j in the neighborhood of node i).

This strategy is the natural extension of FedAvg (the most well-known federated learning approach [1]) to a decentralized setting: the aggregation is performed not by the central controller (as in federated settings) but by each node (and each received model is weighted based on the strength of the social relationship). For this reason, we denote this strategy as DecAvg.

DecAvg is expected to suffer, in a decentralized environment, from the lack of central coordination. In standard FedAvg, the server sends a common initial model to all nodes, which then start the learning with common parameters. Without the common initialization, local models are expected to associate different features with different neurons (due to the permutation invariance of the hidden layers of neural networks). When this happens, coordinate-wise averaging can be detrimental (because nodes are averaging weights that do not match the corresponding learned features). However, the lack of a common initialization can only be solved by either forcing the nodes to coordinate in a decentralized way before the learning phase or by exploiting strategies that do not require coordination. We believe that an initial coordination round is not suitable for dynamic decentralized scenarios (percolating the common initialization across the network might be time-consuming, the nodes of the networks may come and go, etc.). Our research group is currently pursuing the second strategy for its decentralized learning research activities.

3. Experiments

A custom Python simulator [2] has been developed to test decentralized learning strategies in general. The

simulator provides a basic framework, on top of which researchers can plug their decentralized algorithm of choice, as well as define the social network connecting the nodes, either through existing datasets or via well-known network generation models. To test the simulator, we implemented the DecAvg learning strategy described above. Each device is assigned an IID portion of the MNIST dataset [3]. A scale-free synthetic social network with 100 nodes is considered. Each device is equipped with a local AI model (for simplicity, we consider a Multi-Layer Perceptron, MLP). Social-aware gossiping is assessed from the accuracy standpoint, against federated learning (FedAvg [1], specifically) and against a centralized solution where an MLP model is trained at a central server on the whole MNIST dataset. In Figure 1, we observe that the decentralized approach that naively mimics FedAvg suffers from an initial loss of accuracy but catches up with federated learning as the nodes continue to exchange models. At steady state, the decentralized approach and FedAvg are quite close to the performance one would achieve with a centralized approach trained on centralized data. The disruptive initial phase of social AI gossiping is due to the fact that the nodes are not synchronized at the beginning, and each of them has a different initialization of the local MLP model. As anticipated in Section 2 this is the price paid by naive FedAvg-like strategies to the total lack of central coordination.

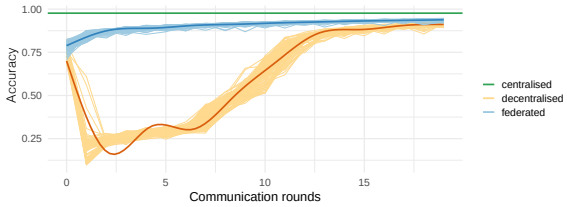

Figure 1: Accuracy of DecAvg vs FedAvg and centralized MLP. The lighter lines represent the accuracy of individual devices, the darker lines are their averages. Centralized learning is only performed at the central server.

4. Current research activities

In the previous section, we have shown how learning strategies that work well for a federated setting might face disruptive challenges in a fully decentralized environment. This calls for innovative approaches that take into account the unique characteristics of a fully decentralized scenario. Our group is currently investigating novel fully decentralized learning strategies that are not affected by destructive averaging. Given the importance of the social graph that connects nodes, we are also studying how learning accuracy and graph topology are linked, as well as which topologies make the learning more or less robust to malicious attacks that aim to interfere with the learning process.

Acknowledgments

This work was partially supported by the H2020 HumaneAI-Net (952026) and by the CHIST-ERA-19-XAI010 SAI project. C. Boldrini and A. Passarella's work was partly funded by the PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR", funded by the European Commission under the NextGeneration EU programme.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, Communication-efficient learning of deep networks from decentralized data, Proceedings of AISTATS 2017 (2017).

[2] L. Valerio, C. Boldrini, A. Passarella, SAI Simulator for Social AI Gossiping, 2021. URL: https://doi.org/10.5281/zenodo.5780042. doi:10.5281/zenodo.5780042.

[3] L. Deng, The MNIST database of handwritten digit images for machine learning research, IEEE Signal Processing Magazine 29 (2012) 141–142.