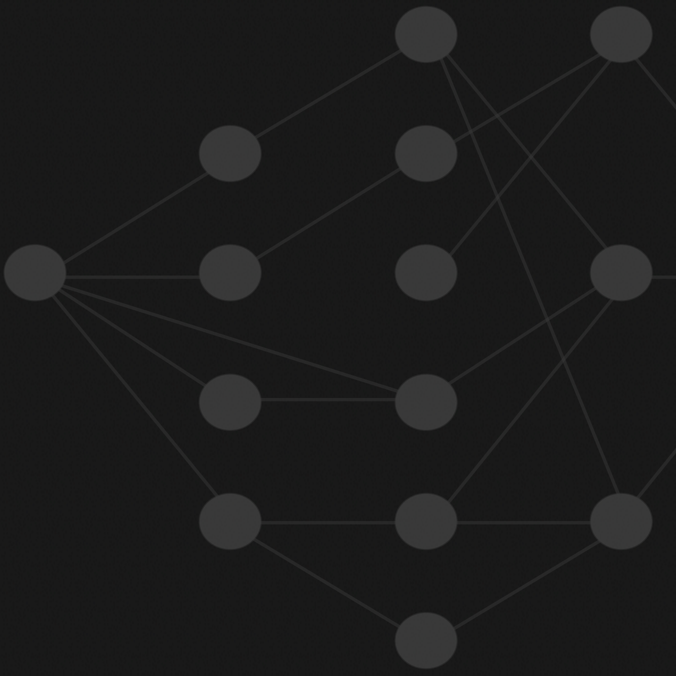
Hyperscalar Research

Federated Learning

Reviewing the state of the art

By:
Hema Krishnamurthy
Research Consultant, Hyperscalar
&
Tony Kenyon
Chief Scientist, Hyperscalar

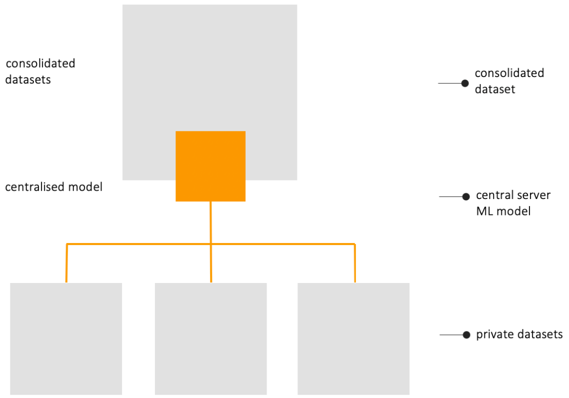


V1.0 August 2020

Contents

01 | Problem statement

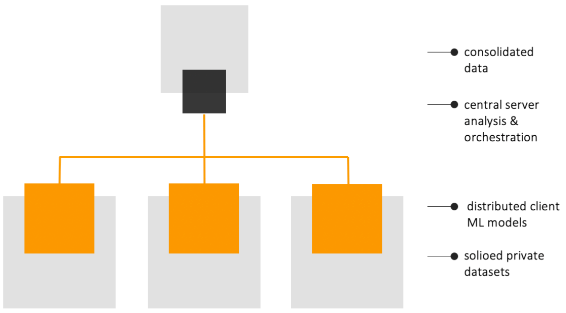
For machine learning models to be trained with acceptable accuracy, it is usually necessary for models be trained on large amounts of high-quality data, from sufficiently diverse sources. This traditionally means gathering large datasets *centrally*, which often proves problematic. Moving large volumes of data can be costly or infeasible, particularly if much of this data is ephemeral, and where it becomes extremely wasteful to replicate into centralised storage.



Arguably the largest concerns here centre on security and privacy; in centralised learning environments, a single central party has complete access to all datasets which means that data owners must implicitly trust the central entity with their data. This is neither ideal, nor strictly necessary.

Data owners often need to retain control to ensure that data doesn't leave their premises - for example, where the data contains sensitive elements such as Personally identifiable Information (PII). There may be concerns around offsite governance due to regulations or policies mandating that data not be moved offsite.

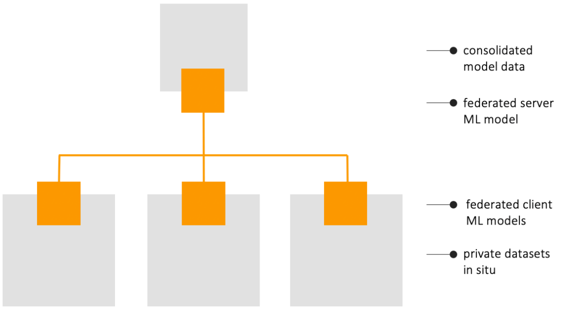
Any restrictions on the ability to move data often leads to *data silos*, with localised learning, perhaps orchestrated through central proprietary integration. This kind of architecture is sub-optimal, especially where valuable insights could be derived by spanning multiple datasets and may lead to higher generalisation errors when models are run on production data. Alternatively, a limited subset of the data may be made available to external parties through APIs for example, or by enabling access to restricted virtual machines. These models can be hard to maintain and run the risk of leaking sensitive information without tight controls on data access queries.


consolidated data
central server analysis & orchestration
distributed client ML models
solioed private datasets

To overcome these challenges, we need a way to efficiently distribute learning, whilst leaving much of the data in situ – only moving what is absolutely necessary to centralised learning models. A new and promising technique has emerged to tackle these challenges, in the form of *federated learning*.

02 |
Enter federated learning

In a federated learning environment, multiple parties retain their own data. Lightweight machine learning models are trained locally on edge devices, and those learnings then transferred back to a global model on the federated server. This means that the federated server only ever sees model updates, and not the raw data. As you might imagine, this kind of model is highly attractive for mobile devices, where users want their data to remain private.



In order to handle potential privacy leaks from these updates, and to preserve privacy of computations, various techniques are typically employed to support federated learning - such as secure multi-party computation (SMPC) and differential privacy.

As an example, IoT devices are a rich source of (often ephemeral) data and hence provide a strong use case for federated learning. With a sharp growth in the number and computational power of IoT devices, it has become an increasingly attractive option to perform edge computing in order to not transmit private information to a central server or cloud.

This also serves to lower the communication overhead of continuously sending large amounts of raw data to the central server.

With federated learning, instead of pushing training data into the central model, the model is brought to the local data, allowing the data custodian to preserve and maintain the single copy of their data. At present federated learning works best with models such as those using iterative gradient descent. Models are run locally on the data, and the weight updates and gradient parameters sent back to the federated server.

The federated server then takes an average of all the updates received, performs a global model update and then shares the results back with the local models. Additional statistical techniques may be used to obfuscate this data further, whilst still preserving overall update integrity.

Multiple rounds of communication take place between the federated server and clients, to train the global model. In each round, the server distributes the global model updates out to a subset of clients.

The clients perform local training, and to minimize communication overhead, clients might take several steps of mini-batch gradient descent during a single communication round. Alternatively, communication overhead reduction can also be achieved via *gradient sparsification* (GS) techniques, where only a subset of the gradients is sent back to the federated server[5].

Next, the optimised models are sent back to the federated server and aggregated (e.g. by averaging), to update the global model.

Depending on the performance of the new global model, training may be terminated (i.e. the model may be frozen if it performs suitably well), or a new communication round starts (next round of model update begins).

From the discussion so far, it should be clear that federated learning requires quite a sophisticated architecture, may not be suitable for all learning models, and throws out a number of interesting and unique challenges.

03 | Challenges

Even though private training data never leaves the local data source, federated learning is complex to orchestrate and not without its challenges – these systems typically require tuning to gain optimal performance and reliability given the complexity of the network infrastructure and synchronisation needs. Additional challenges are highlighted below:

Legitimacy and computational power of clients

To ensure only genuine devices participate in the network, identity verification/authentication is necessary to ensure that malicious parties do not contribute to the model updates and poison the upstream global model. Reputation based systems can be put in place to ensure clients are contributing only clean data to the local model updates. Since the computational resources of edge devices can vary significantly, this may also affect the performance of the local training.

Synchronisation challenges

Typically, federated learning has a high communication overhead. This is partially solved by the local training/updates (which reduces the number of back and forth messages between the federated server and clients) and compression of the models being sent to the federated server. However, traditional compression techniques may need to be adapted for model compression. Model convergence times may also be different on different clients and this adds an element of unpredictability to the training process.

Tuning these models to operate efficiently and reliably gets complicated when synchronising hundreds of thousands of updates.

Data generation

While one of the motivations for federated learning is the use of diverse data from different devices, model training complexity could grow out of bounds because typically the training data needs to satisfy the i.i.d (independent identical distribution) assumptions i.e. that the training data needs to be *independent* from the test data, but should have an *identical distribution* (probability distribution) to the test set. It becomes increasingly difficult to enforce this on raw data that comes from heterogenous devices.

Moving beyond Supervised learning

Supervised learning works on labelled data. In a federated model, it becomes important that the data between different clients be labelled in a consistent manner for the global model updates to yield useful results. Making this operate consistently and reliably when the number of devices can be potentially very large, and the devices may be heterogenous becomes challenging.

The availability of well labelled data is becoming a major challenge in machine learning as data volumes increase, especially in real-time contexts. In federated environments - such as mobile networks – it may be infeasible to label data using offline methods. We may for example see data being labelled dynamically, based on user behaviour and interaction with the client application.

Unsupervised and semi-supervised learning works with unlabelled (or partially unlabelled) data. Further research is needed to understand the how efficiently a federated setting works for unsupervised models. There is also increased interest in self supervised labelling techniques to assist in dynamic labelling and avoid reliance on domain expertise.

Security and privacy

As we have seen, in federated learning systems the training process is handled on local devices and only the model updates shared with the central server. However, there are still several types of attacks that can be performed against such systems.

Active attacks - where 1) malicious clients can poison the local data/inputs to the training model to affect the training result, or 2) a malicious federated server can tamper with the training process if the model updates are in the clear.

Passive attacks - where adversaries can observe and learn from the gradient updates and potentially reverse engineer sensitive source information.

Therefore, in a federated environment, we must consider privacy on the *computations* as well as the *outputs*. Technologies like secure multi-party computation and homomorphic encryption (HE) offer high *computational* security. Differential Privacy (wherein noise is added to the raw data to obfuscate) provides a mathematically justified approach to guarantee *output* privacy. [2,3]

That said, sophisticated cryptographic techniques using SMPC can add to computational latency and use of local Differential Privacy (where data is effectively modified) could affect the model performance. Hence the trade-offs between security/privacy and model performance needs to be carefully balanced.

Auditability

Ideally, the behaviour of the federated clients and server should be auditable so they can be independently verified - particularly where we need the model to be 'explainable' or we need to investigate why a model made decisions in the event of bias [3,4]. Distributed auditing could be achieved via the use of blockchains or bulletin boards, for example before each client sends out a model update, it could also post a commitment to the bulletin board or blockchain, and the recipient (i.e. the federated server) only accepts the update if it matches the commitment [5].

04 | Conclusion

Today we see a number of mature tools and frameworks (e.g. Snowflake, Amazon SageMaker) which make the task of moving and analysing data centrally in the cloud relatively easy. It remains to be seen whether cloud-based learning, together with improved governance processes and increased cost reduction will remain dominant.

Federated learning offers a potentially attractive way forward by decoupling data from purely centralised models, pushing learning out to the edge. For organisations looking to monetise sensitive proprietary data this achieves many of the benefits of centralised learning, whilst retaining security and privacy of client data, with the cost savings of not having to move data.

That said, whilst collaborative learning in a federated setting provides a attractive means to train models on the edge, it is not without its challenges - it introduces significant additional complexity, and is currently not suitable for all learning models and applications.

Federated learning is a fascinating but nascent field, and more research needs to be done to mitigate some of the challenges outlined, in order to extend the scope of decentralisation with machine learning. For now, it is prudent to carefully evaluate whether a traditional centralised learning approach, with complementary security and privacy techniques will be sufficient, or whether there is widespread demand for federated techniques.

References

1. A hybrid approach to privacy preserving federated learning. See: https://arxiv.org/pdf/1812.03224.pdf

2. The algorithmic foundations of differential privacy: See: https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf

3. Explainable machine learning in deployment. See: https://arxiv.org/pdf/1909.06342.pdf

4. See: https://hbr.org/2019/10/we-need-ai-that-is-explainable-auditable-and-transparent

4. FLChain: A Blockchain for Auditable Federated Learning with Trust and Incentive. See: https://www.researchgate.net/publication/337526695_FLChain_A_Blockchain_for_Auditable_Federated_Learning_with_Trust_and_Incentive

5.See: https://researcher.watson.ibm.com/researcher/files/us-wangshiq/PH_ICDCS2020.pdf

About Hyperscalar

Hyperscalar provides research and advisory and IP diligence functions, specialising in private equity, as well as advising and mentoring start-ups in disruptive technologies such as machine learning, blockchain, process automation, cybersecurity and robotics. For further information see **www.hyperscalar.com**