

Swarm Learning - A Fully Decentralised Approach To Machine Learning

Josh Pattman

March 2023

University Of Southampton

The Problems

Privacy

- Data stored in multiple locations
- Cannot share the data between locations for privacy reasons
- *Medical records*

Performance

- Machine learning needs lots of processing power
- A supercomputer is not available to many
- However they may have access to many lower power devices (nodes)
- *Company with many unused computers during the night*

Federated Learning - The Current Solution

- A single model is stored on the server
- Each node has its own dataset
 - This is not shared with other nodes or the server
- The model can be shared between the server and clients
- **Goal:** Perform machine learning without sharing the data

Federated Learning - How Does It Work?

- Many variations of federated learning
 - One of the originals is *Federated Averaging* (*FedAvg*)
 - Many other algorithms are based off this
- FedAvg has repeated training steps. Each step:
 1. Server sends model to a set of nodes
 2. Nodes perform training on the model
 3. Nodes send their models back to server
 4. New model is the average (mean) of all nodes models

Federated Learning - Issues

- Vulnerable to central server going down
- Requires that every node has direct access to the server
- Few slow nodes slow the whole process down

Swarm Learning

- Each node has a distinct model, called the *local model*
 - Every model approximates the *global model*
- Each node has its own dataset
 - This dataset cannot be shared with any other nodes
- The goal is to train the *global model* using all available data
- No central server or node acting as a central server

Swarm Learning - How Does It Work?

- Repeated Training Steps. Each node each step:
 1. Perform training on the local model
 2. Send trained model to all neighbours
 - This will get cached on the neighbour
 3. New local model is the combination of all neighbours most recent local models
- During step 3, the cached models are used to prevent the node having to wait for responses

Swarm Learning - Specifics

- Different combination methods
 - Combine by average
 - Combine with learning rate
- Only combine neighbours who have done more training than this node
- Wait for certain number of neighbours to catch up

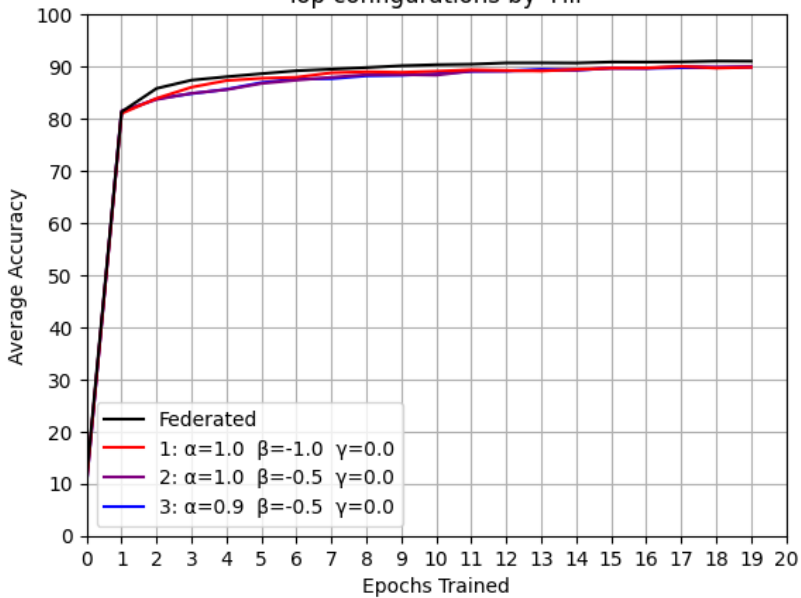
Swarm Learning vs Issues of Federated Learning

- ~~Vulnerable to central server going down~~
 - No central server - to stop training you would have to take out every node
- ~~Requires that every node has direct access to the server~~
 - Swarm learning can function on sparse networks of nodes
- ~~Few slow nodes slow the whole process down~~
 - You never have to wait for a node due to caching

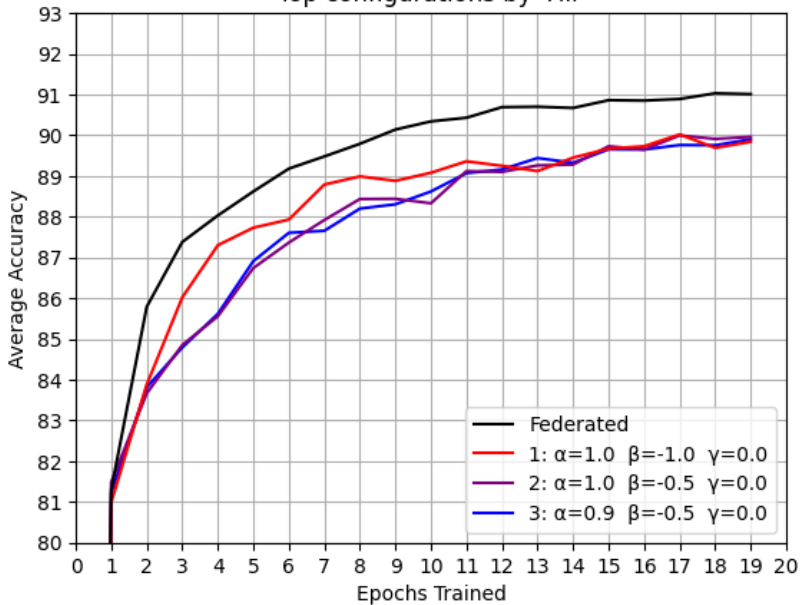
Swarm Learning - Performance

- Many different configurations of the algorithm, can drastically affect performance
 - In following plots only top 3 in a category have been shown
 - For example may find top 3 configurations by 'area under graph'
- Following plots are *accuracy* of classifying MNIST Fashion, and x axis is number of epochs trained
 - To make the problem a little harder each node only has 10 percent of the dataset
- Federated Averaging is also shown

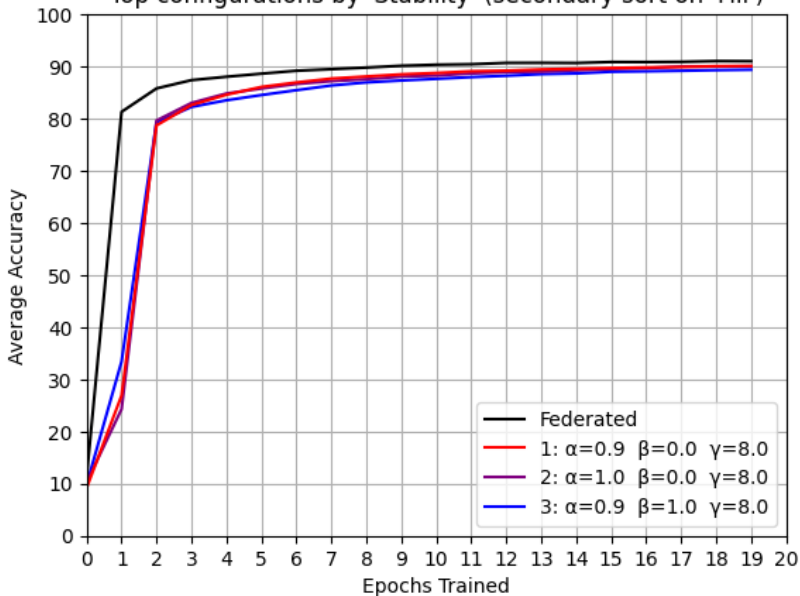
Top configurations by 'Fill'



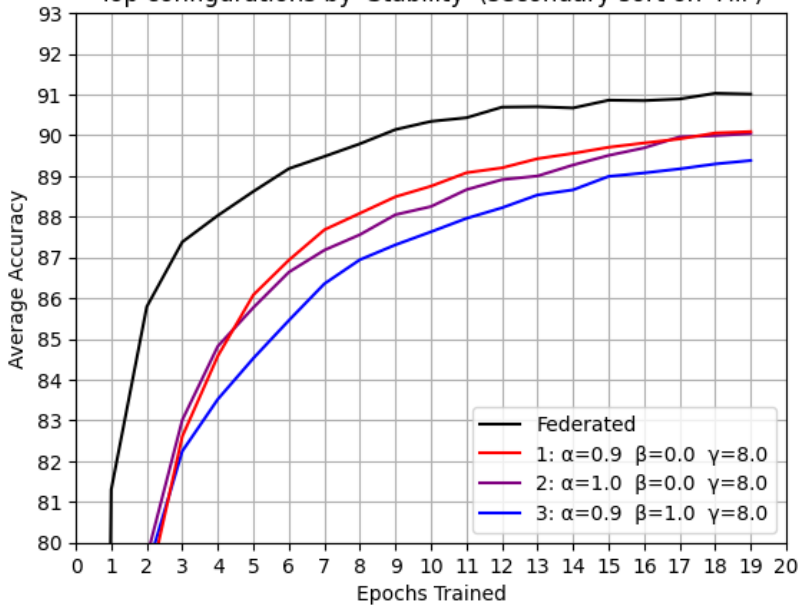
Top configurations by 'Fill'



Top configurations by 'Stability' (secondary sort on 'Fill')



Top configurations by 'Stability' (secondary sort on 'Fill')



- Those plots were in a densely connected situation
 - I have not got round to testing sparse networks yet
- It would also be ideal to test more datasets but I don't think I will be able to due to time constraints

Conclusion

- Swarm Learning is a promising machine learning algorithm for training a model on data distributed on private data islands
- It addresses some of the issues with Federated Averaging, one of the current techniques
- It does not perform quite as well as Federated Averaging in a densely connected network

Thanks for listening!

- LinkedIn: [**linkedin.com/in/josh-pattman**](https://www.linkedin.com/in/josh-pattman)
- GitHub: [**github.com/JoshPattman**](https://github.com/JoshPattman)