

# Swarm Learning - A Fully Decentralised Approach To Machine Learning

---

Josh Pattman

March 2023

University Of Southampton

# Why Distributed Machine Learning

---

# 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

---

# Federated Learning - The Current Solution

- A single model is stored on the server
- Server controls many nodes - computers that can perform training
- Each node has its own dataset
  - This is not shared with other nodes or the server
- **Goal:** Perform machine learning by only sharing the model, not the data

# Federated Learning - Variations

- Many variations of federated learning
  - One of the originals is *Federated Averaging* (*FedAvg*)
  - Many other algorithms are based off this

# Federated Learning - How Does It Work?

- 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

---

# Swarm Learning

- No central server/node
- Each node has a distinct model, called the *local model*
  - Must keep all local models close to each other
- Each node has its own dataset
  - This dataset cannot be shared with any other nodes
- The goal is to train all local models using all available data

# 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 saved on the neighbour
  3. New local model is the combination of all neighbours most recent local models

# 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 with this node

# 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 to communicate, instead use the most recent model it sent

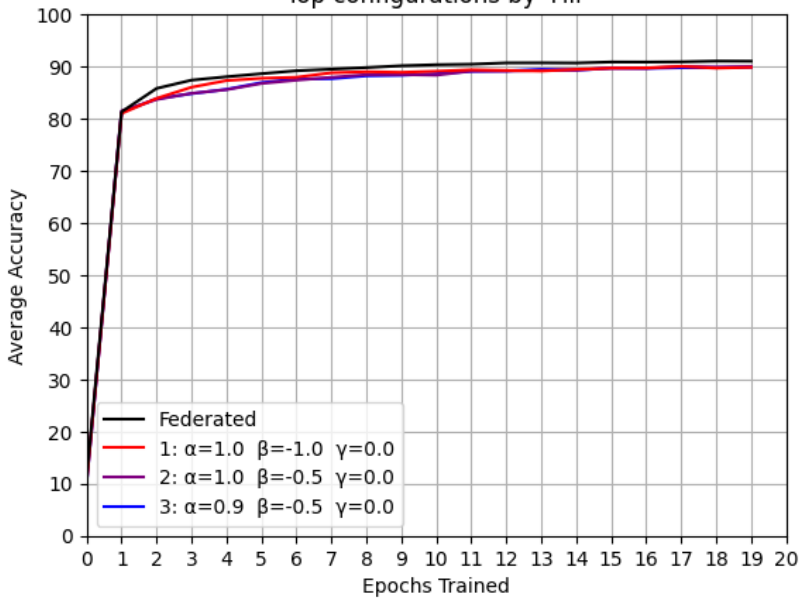
# Results

---

# Swarm Learning - Performance

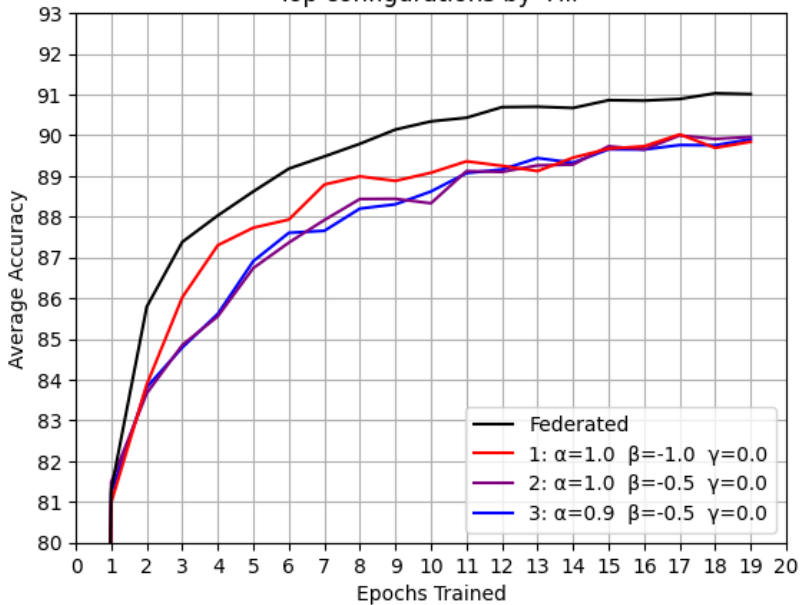
- Following plots are *accuracy* of classifying MNIST Fashion, and x axis is number of epochs trained
  - To make the problem a little harder each node only has 10 percent of the dataset
- Many different configurations of the algorithm, can drastically affect performance
  - In following plots only top 3 configurations sorted by a specific metric
  - For example metric might be 'area under graph'
- Following plots nodes are densely connected

Top configurations by 'Fill'

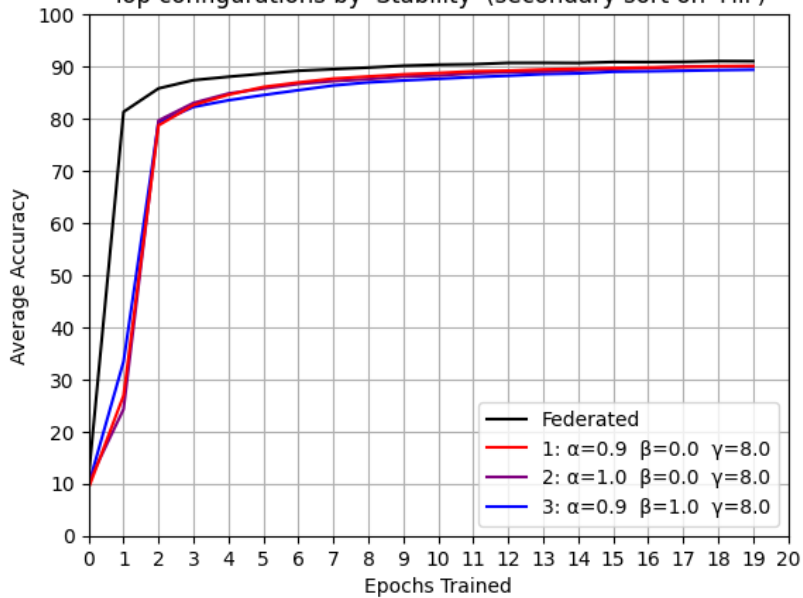




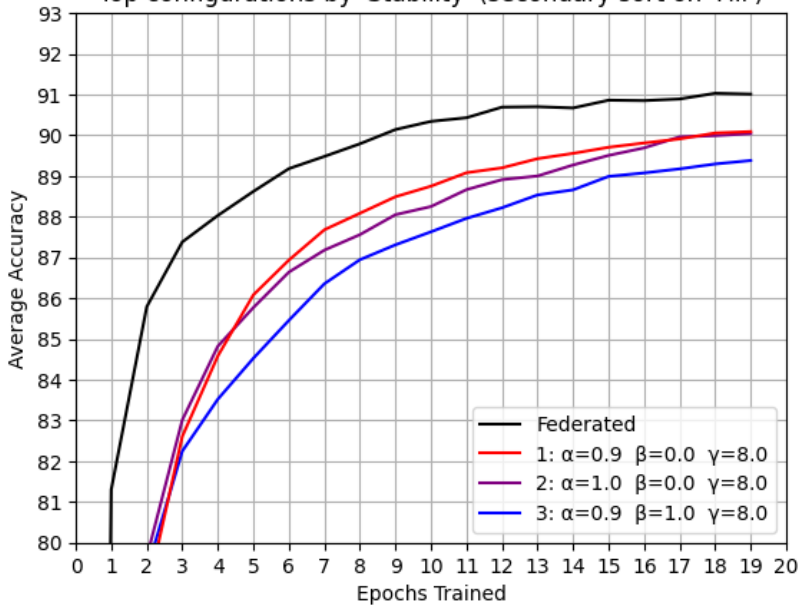
Top configurations by 'Fill'



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



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



# 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)