Optimizing Federated Learning Hyper-Parameters in Flower
What is Federated Learning?

- Federated Learning is a form of distributed Machine Learning working over large numbers of resource constrained devices.
- It attempts to reduce communication costs by alternating local training with global aggregation of model parameters without ever directly sharing client data.
Federated Learning At A Glance

Global Server

Global Model

Client

Client
Federated Learning At A Glance

Global Server

Train for E epochs

Client

Train for E epochs

Client
Federated Learning At A Glance

Global Server

Local Models

Client

Client
Federated Learning At A Glance

Global Server

Local Models

Client

Client
Federated Learning At A Glance

Global Server

Model Aggregation

Client

Client
Flower is a Federated Learning framework meant to allow full FL model development and deployment.

- It offers a variety of aggregation algorithms and options for customization.
- What sets it apart?
Flower is a Federated Learning framework meant to allow full FL model development and deployment.

- It offers a variety of aggregation algorithms and options for customization.
- What sets it apart?

<table>
<thead>
<tr>
<th></th>
<th>TFF</th>
<th>PySyft</th>
<th>LEAF</th>
<th>Flower</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterogeneous clients</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Scalability</td>
<td>*</td>
<td>(✓)**</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Server-side definitions</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ML framework-agnostic</td>
<td>***</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Language-agnostic</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Baselines</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

planned * only Python-based instances ** limited to PyTorch and TF/Keras ***
Problems in Federated Learning

Data Heterogeneity:
- Unlike other ML settings, client data may be highly non-iid
- This can cause the entire global model to diverge
- Requires controlling the potential impact of a client model as well as the amount of training performed

System Heterogeneity:
- Device types may vary wildly in terms of specifications and internet connection
- Makes training unreliable

Global-Local Accuracy Trade-Off:
- Clients with unusual data and system characteristics end up with a worse model than they could have trained on their own
Hyper-Parameter Optimization Component For Flower

Motivation:
- Handling data and system heterogeneity requires tuning both the aggregation algorithm and the clients
- Determining how much local computation should be done and with what parameters
- Flower currently lacks any means of optimizing parameters over the federated network

Method:
- Bayesian Optimization with BoTorch---subject to change---is the proposed optimization method, given that training and evaluating a federated network takes quite a long time.
- The component should be able to handle FL algorithms as black-boxes and allow the user to specify which parameters to optimize. Most classic FL aggregation algorithms are controlled by a few, less than 10, main parameters
- After construction, performance of the baseline models provided by flower will be evaluated after optimization
Justification and Impact

**Justification:**
- My MPhil dissertation focuses on **local adaptation** methods applied to Federated Learning, to be implemented in Flower.
- Hyper-parameter optimization represents a parallel research direction which may help inform my work.
- Most of the current Federated Learning work focuses on constructing increasingly complex aggregation algorithms, it is worth investigating if auto-tuning older algorithms is sufficient for them to compete on the baseline tasks.

**Impact:**
- Flower is intended to be the primary FL framework in the vein of Tensorflow/Pytorch for general machine learning.
- Constructing a new component for Flower, if accepted by the team, would represent a direct contribution to the FL community for both research and final-product development.
- Hyper-parameter optimization has only began being explored in FL recently, one paper from 2019 and one from 2021, and providing more data for it would be directly useful.
Questions?