Flow: Fine-grained Personalized Federated Learning through Dynamic Routing

Kunjal Panchal (kpanchal@umass.edu) and Hui Guan
University of Massachusetts, Amherst

① Introduction

Statistical heterogeneity can be detrimental for clients in FL
- need for **personalization**

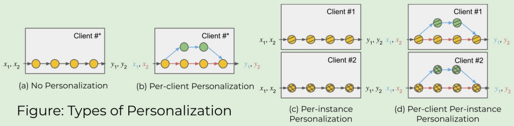


Figure: Types of Personalization

Existing approaches are per-client
Need for a **per-instance** approach:
- instances that fall under the global data distribution can benefit more by using the global model

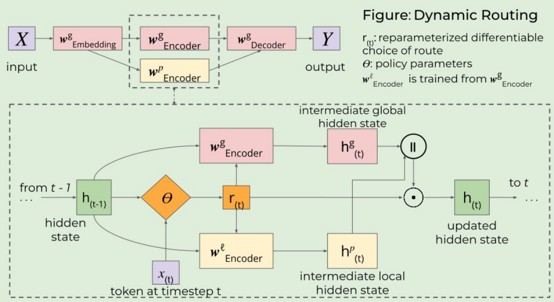
③ Inference & Experiments

A client doesn't persist any personalized/local states, since it's impractical in cross-device FL
(1) Generates w^ℓ from w^g
(2) Trains routing policy θ on latest feature repr learned by w^g

We validate our approach against:
- No personalization: to understand w^g behavior
- Per-client personalization: to see how much w^ℓ is better over w^g
- Stateful personalization: to observe the disadvantages of clients without w^ℓ or with stale w^ℓ

② Methodology

Depending on the input it receives, Flow creates a personalized model that can dynamically make decisions on
(1) when to use a client's local parameters, and
(2) when to use the global parameters.



Stages of **Flow**:
(a) Receive the global model w^g. Create a local model w^ℓ from w^g by finetuning for single epoch.
(b) Create the personalized model which has w^g, w^ℓ, and a dynamic policy θ for per-instance routing.

Shallow sequential models are computationally less intensive, and still allow dynamic routing temporally, hence we pick language tasks
- $r_t = \text{softmax}_\tau(f_\theta([x_t; h_{(t-1)}]))$ *[prob. of picking routes w^g and w^ℓ]*
- $h_t = [g(w^g, h_{(t-1)}, x_t);\ \ell(w^\ell, h_{(t-1)}, x_t)] \cdot r_t$ *[updating hidden state with the mixture of global and local updates, based on the route probability]*

▪ Beneficial for the newly joined clients with heterogeneous data, which has insufficient sample count for a near-optimal w^ℓ.
▪ This improves inference performance of those instances which are not well-learned by the less generalizable local model.

References

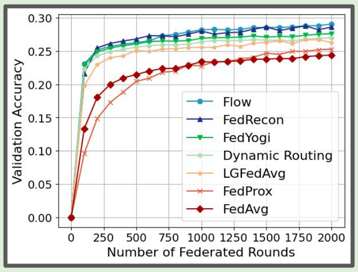
1. Federated Reconstruction: Partially Local Federated Learning by Karan Singhal et al *NeurIPS 2021*
2. Neural Speed Reading via Skim-RNN by Minjoon Seo et al *ICLR 2018*
3. Federated Optimization in Heterogeneous Networks by Tian Li et al *MLSys 2020*

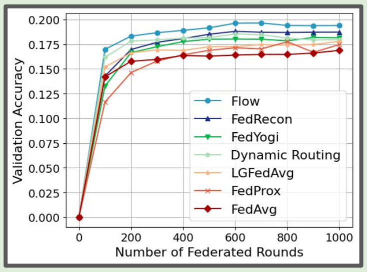
④ Results

Baseline	Stackoverflow	Reddit
FedAvg ◇	24.98%	17.30%
FedYogi ◇	27.90%	18.85%
APFL † §	25.48%	18.48%
Ditto † §	25.55%	18.74%
FedProx ‡ §	25.38%	17.58%
LG-FedAvg ‡ §	26.78%	18.59%
FedRecon (500 OOV) ‡ §	28.26%	18.89%
Dynamic Routing ‡ § ♭	26.54%	18.70%
Flow ‡ § ♭	**29.16%**	**19.36%**

◇ = Server-side Optimization
† = Stateful Pers.
‡ = Stateless Pers.
§ = Per-Client Pers.
♭ = Per-Instance Personalization

Validation (below) and Test (left) accuracies for Reddit and Stackoverflow datasets respectively





⑤ Discussion

- FedYogi performs better than client-side personalization FedProx and LGFedAvg (both uses FedAvg as server-side aggregation) because of **low statistical variance** across clients
- Per-instance Dynamic Routing outperforms per-client FedProx and LGFedAvg, proving that we can **seamlessly integrate local** model with the **global** model, to outperform individual models
- FedRecon (using FedYogi as aggregation), **outperforms stateful** approaches since the latter require training of personalized models over multiple rounds, which is impractical when only ~5-6% of total clients participate
- Flow outperforms FedRecon since it is **finer-grained** in its choice of a client-specific local model and a generalized global model