System Design

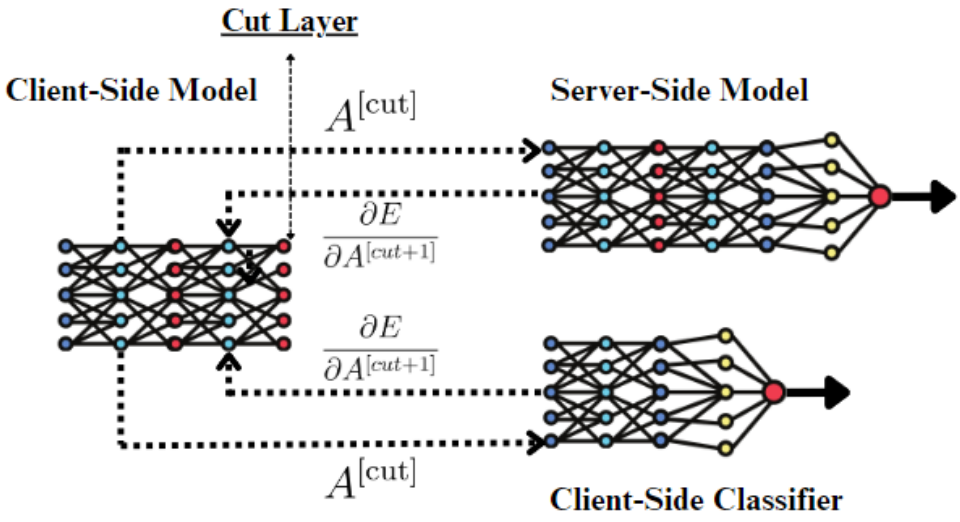
FL Task

An FL task is a specific computation for an FL population, such as training to be performed with given hyperparameters, or evaluation of trained models on local device data.

- Image classification or Object Detection
- Must agree on a homogenous model architecture (all clients share the same model architecture)
- Must agree on a common predetermined set of global tasks (possible outputs). (The union of all the unique local tasks)
- Both clients and the server have predictive capabilities
- Clients do not have labels, only input samples → Server provides supervision signal
- Employ FL and SL

Model Architecture

Since we want both parties to have predictive capabilities and implement SL, our network must be a multi-exit neural network. We will split our initial model into two parts (client-side and server-side) and allocate them to their respective party. However, due to the model splitting, the client does not have an output layer, thus we must add a classifier (some FC layers) resulting in a multi-exit neural network.



Where $A^{[cut]}$ is the outplut of the cut-layer and $\frac{\partial E}{\partial A^{[cut]+1}}$ is the partial derivative of the error with respect to the classifier's/server's last layer.

Let u be the multi-exit model: $u = [\phi, h, \theta]$
where ϕ and h are the client-side model and classifier respectively and θ is the server-side model.

Therefore, each client k has the multi-exit model $u_k = [\phi_k, h_k, \theta]$

Important

All clients must have the same number of output nodes. (One node for each unique local task within the network)

Data Availability

Cross-Silo FL: Few clients (approx. 100, usually indexed), high availability for training and reliable network.

Network Topology

Centralized FL

- centralized server coordinates the process
- data remain decentralized on the edge

Data Partition

Horizontal FL (similar feature space among clients' data but different sample space)

Training Protocol

1. Client forwards input up to the cut-layer and transmits the produced features to the server
2. Server receives and forwards the features then creates and transmits the supervision signal
3. Client receives the signal, ensues the forward-propagation from the cut-layer to its classifier and uses the received signal as a ground truth.

FL Protocol

Server-Side

1. Server transmits initial weights to participants
2. For each client sequentially do:
    a. For each batch of data:
        i. Receive features
        ii. Forward Propagate
        iii. Create supervision signal
        iv. Transmit signal to client
    b. Receive client's updated weights.
3. Aggregate global model

Client-Side

1. Receive initial weights
2. For each batch of data:
    a. Forward-feed up to the cut-layer
    b. Transmit cut-layer's features

c. Receive supervision signal

d. Forward-feed from the cut-layer up to the output layer

e. Calculate error using the supervision signal

f. Update model's parameters

3. Transmit updated weights

System Failures

Identify potential issues hindering training and decide on how to tackle them.

Network Issues:

1. Client misses initial weights
2. Server misses a packet when receiving features
3. Client misses supervision signal
4. Server misses uploaded weights

Model issues

1. Either party's model experiences an issue during training.

Log events for every state in a training round and use ASCII characters to depict each state.

Can analyze data, distinguish issues, verify solutions and visualize the processes.

Session Shape	Count	Percent
-v[]+^	1,116,401	75%
-v[]+#	327,478	22%
-v[!	29,771	2%

Table 1: Distribution of on-device training round sessions. Legend: - = FL server checkin, v = downloaded plan, [= training started,] = training completed, + = upload started, ^ = upload completed, # = upload rejected, ! = interrupted.

In all failure cases the system will continue to make progress, either by completing the current round or restarting from the results of the previously committed round.

Questions

- If the server can produce the labels, thus its predictions are considered ground truth, what is it training for? Do we train both sides of the model or just the client? (to deduce the appropriate cost function)
- Ο ρόλος του server πέρα από τον συντονισμό της εκπαίδευσης είναι να παράγει μόνο το σήμα μάθησης ή και να χρησιμοποιείται για predictions?

Possible Problems

- In order for the server to be able to infer from a client's feature maps, these feature maps must be 'meaningful' enough to gauge his prediction. Even if the server is able to correctly classify a sample given the raw data, when inferring from feature maps they may not always point him towards the right direction. (If the model is pre-trained and is relatively large then the first layers may be good feature extractors)

FL System Design Papers:

Architectural Patterns for the Design of Federated Learning Systems *(Lo et al., 2021)*

Towards Federated Learning at Scale: System Design *(Bonawitz et al., 2019)*

Industrial Federated Learning – Requirements and System Design *(Hiess et al., 2020)*

A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection *(Li et al., 2023)*