Project Report

Federated Learning in IoT–Fog–Cloud Hierarchies Using Progressively Deepening Split Neural Networks

Submitted by

Pranav Gupta 22JE0710
Saransh Shivhare 22JE0867
Ayush Shaw 22JE0221

Under the guidance of

Prof. Prasanta K. Jana
Department of Computer Science and Engineering

Table of Contents

Contents

Federated Learning in IoT–Fog–Cloud Hierarchies Using Progressively Deepening Split Neural Networks

Abstract

Federated Learning (FL) has emerged as a paradigm-shifting approach for privacy-preserving machine learning, especially in the context of the Internet of Things (IoT). However, traditional FL frameworks are ill-suited for the complex, hierarchical, and resource-heterogeneous reality of modern IoT-Fog-Cloud ecosystems. These frameworks suffer from significant limitations, including high communication overhead, poor scalability, and an inability to accommodate the severe resource constraints of edge devices (clients). Furthermore, the statistical heterogeneity (Non-IID data) of devices and the "one-size-fits-all" model architecture lead to unstable convergence and context loss.

This project introduces a novel framework: **Progressive Multi-Tier Neural Network Construction for Federated Learning**. Our approach fundamentally redesigns the learning process by creating a hybrid Split-Federated Learning (SFL) model that aligns with the physical IoT-Fog-Cloud hierarchy. In our architecture, the neural network model itself is progressively "grown" across the tiers. Lightweight feature extractors (f_{ω_i}) are trained on resource-constrained IoT devices, intermediate head layers (h_{ϕ_j}) for regional context are trained on fog nodes, and deep inference layers (g_θ) for global aggregation are trained on the cloud.

This progressive construction allows IoT devices to participate with minimal computational load (approaching zero-compute for simple forward-pass clients), while fog nodes act as intelligent intermediaries that handle regional aggregation and context-aware learning. We present the system architecture, the mathematical formulation for this multi-tier composite model, and the detailed training workflow that combines split execution with hierarchical federated aggregation. The framework is designed to reduce end-to-end latency, minimize cloud-bound communication, and improve model robustness to Non-IID data, thereby enabling scalable and efficient deep learning in large-scale IoT deployments.

Keywords: Federated Learning, Split Learning, IoT, Fog Computing, Deep Neural Networks, Progressive Architectures

1 Introduction

With the rapid expansion of IoT networks, the data generated at the edge of the network has increased dramatically. Traditional centralized deep learning models struggle with privacy, latency, and communication bottlenecks. Federated Learning (FL) distributes the training process but assumes sufficient computational capacity at all clients, which is unrealistic for many IoT devices.

To overcome these challenges, this work proposes a **Progressively Growing Neural Network (PGNN)** framework embedded within a **Hybrid Split Federated Learning (HSFL)** paradigm. The approach distributes a neural network hierarchically across **IoT**, **Fog**, and **Cloud** tiers, where model depth and abstraction progressively increase from the edge to the cloud.

2 Literature Review

This section reviews existing works on **Federated Learning (FL)**, **Fog-based Federated Learning (FML)**, **Split Neural Networks (SplitNN)**, and **Neural Network architectures**, providing the conceptual foundation for the proposed progressively growing hybrid approach.

2.1 Federated Learning (FL)

Federated Learning introduced the **FedAvg** algorithm to enable distributed model training without centralizing data. Each client performs local updates and transmits weight parameters to a central server for aggregation. Although FL maintains data privacy, it assumes moderate computational power at each client, limiting scalability for low-resource IoT devices.

2.2 Fog-based Federated Learning (FML)

To address scalability and latency, Fog-based or Hierarchical Federated Learning architectures were proposed. These architectures introduce **intermediate fog nodes** that aggregate models from nearby clients before global synchronization at the cloud. However, most FML implementations still rely on full local model training at clients, which remains infeasible for ultra-light IoT nodes.

2.3 Split Neural Networks (SplitNN)

SplitNN, proposed by Gupta and Raskar, introduced the idea of **splitting a deep neural network** between clients and servers to minimize client computation. Nevertheless, existing SplitNN architectures are mostly two-tier (client–server) and lack dynamic hierarchical expansion across fog and cloud layers.

2.4 Neural Network Architectures and Progressive Deepening

Classical deep learning models such as AlexNet, VGG and ResNet demonstrate that deeper architectures improve hierarchical feature extraction and abstraction.

3 Problem Statement

The objective is to design a hierarchical, resource-adaptive learning system that:

- Enables collaborative model training among IoT, Fog, and Cloud nodes.
- Minimizes client-side computation and communication cost.
- Preserves data privacy while achieving high accuracy.
- Supports progressive deepening of the neural network across layers.

4 Proposed Architecture

4.1 Hierarchical Design

The proposed framework consists of three computational tiers:

1. IoT Layer (Client Level):

- Hosts lightweight feature extractor modules $f_{\omega_i}(x)$.
- Performs only forward propagation to compute smashed activations a_i.
- Transmits a_i to the fog node with minimal computation and power usage.

2. Fog Layer (Intermediate Level):

- Receives a_i from multiple IoT nodes.
- Adds intermediate head layers h_{ϕ_j} to learn regional representations.
- Performs local updates and aggregates weights from its IoT clients.
- Sends refined features r_j to the cloud for global training.

3. Cloud Layer (Global Level):

- Adds final inference layers g_θ for high-level decision-making.
- Aggregates weights across fog nodes to form a global model.
- Sends updated parameters back down for local adaptation.

4.2 Progressive Network Growth

The model deepens hierarchically:

$$f_{\omega_i} \;\rightarrow\; h_{\phi_j} \;\rightarrow\; g_\theta$$

Each higher tier appends new layers to enhance representational capacity (Fig 1). IoT devices handle shallow feature extraction, fog nodes generalize local patterns, and the cloud captures global knowledge. This leads to efficient usage of higher resource capacity of fog and cloud layers.

5 Training Workflow

1. **Forward Pass:** IoT clients compute activations $a_i = f_{\omega_i}(x)$ and send them to fog nodes.
2. **Fog Processing:** Fog nodes compute $r_j = h_{\phi_j}(a_i)$, perform backpropagation, and update local parameters.
3. **Cloud Aggregation:** The cloud aggregates regional updates and performs global optimization:

$$\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}_{cloud}$$

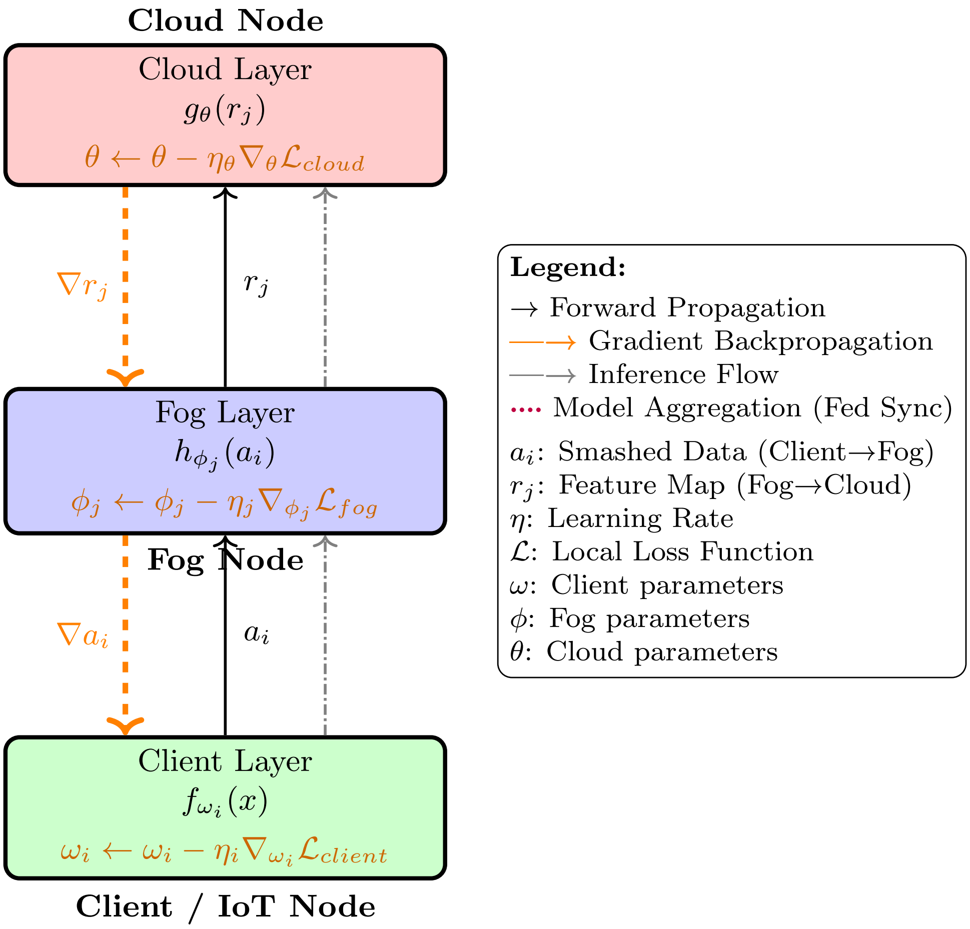

Figure 1: Proposed progressively growing neural network across IoT–Fog–Cloud layers.

4. **Downward Synchronization:** Updated parameters are propagated back to fog and IoT nodes.

5. **Inference:** Final inference combines all layers:

$$\hat{y} = g_\theta(h_{\phi_j}(f_{\omega_i}(x)))$$

6 Planned Implementation Details

Key Configurations

Aspect	Configuration / Suggestion
Frameworks	PyTorch + Flower/FedML (for orchestration), PySyft (for SplitNN simulation), MQTT/gRPC (for IoT–Fog communication).
Datasets	CIFAR-10 / Tiny ImageNet (vision), Edge-IIoTset (industrial IoT), UCI-HAR (sensor classification).
Model Partitioning	IoT: 2–3 conv layers; Fog: mid-level heads; Cloud: fully connected + inference layers.
Training Setup	IoT performs forward-only; Fog performs full training; Cloud aggregates weights every 10–15 epochs. Optimizer: Adam, learning rate 1×10^{-3}.
Metrics	Accuracy, Communication Cost, Latency, Convergence Speed, Energy Consumption.
Baselines	FedAvg, SplitNN, Hierarchical FL.
Expected Gains	$\downarrow$ 60% client computation, $\downarrow$ 40–50% bandwidth, $\downarrow$ 30% latency, comparable or improved accuracy.

7 Advantages

- **Compute Efficiency:** Heavy training offloaded to fog and cloud layers.
- **Communication Reduction:** Only activations and gradients are exchanged.
- **Low Latency:** Fog-level proximity reduces round-trip delay.
- **Scalability:** Supports heterogeneous devices with adaptive depth allocation.
- **Privacy:** Raw data remains at the client level.
- **Progressive Adaptability:** Model deepens dynamically across layers.

8 Novelty and Contributions

1. Introduction of a **progressively growing neural network** spanning IoT–Fog–Cloud tiers.
2. Development of a **Hybrid Split-Federated Learning scheme** combining split computation and hierarchical aggregation.
3. Enables participation of **zero-compute clients**, expanding FL inclusivity.
4. Multi-tier aggregation reduces bandwidth while maintaining global accuracy.
5. Dynamic allocation of layers and training responsibility per device capability.

9 Future Work

- Adaptive layer partitioning using reinforcement learning.
- Incorporation of quantization and pruning for energy optimization.
- Integration of secure aggregation and homomorphic encryption.
- Validation across multiple domains (vision, NLP, sensor analytics).

10 Conclusion

The proposed **Progressively Growing Neural Network in a Hybrid Split Federated Framework** introduces a scalable and resource-aware approach for distributed intelligence in IoT–Fog–Cloud ecosystems. By enabling hierarchical model expansion, it minimizes client load, reduces latency and communication, and preserves data privacy while ensuring high global accuracy — forming a strong foundation for next-generation federated IoT intelligence.