Real-time, robust, and reliable (R^3) machine learning across wireless networks

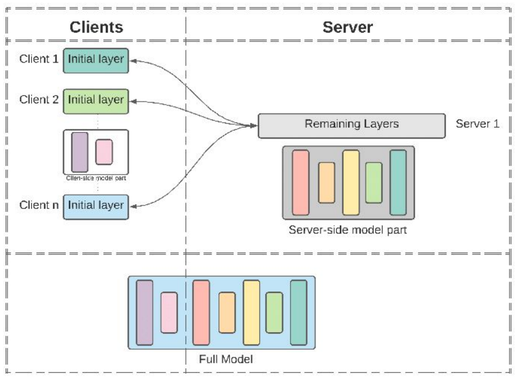
Akshar Vedantham, Varun Kota, Kirthana Ram
Advisors: Professor Anand Sarwate, Professor Waheed Bajwa

RUTGERS
WINLAB | Wireless Information Network Laboratory

SCAN ME

What is Split Computing?

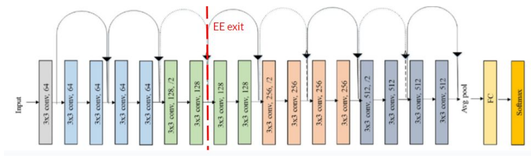
- Divides neural networks into parts for execution on different devices.
- Runs one part on a mobile device and the other on a powerful server.



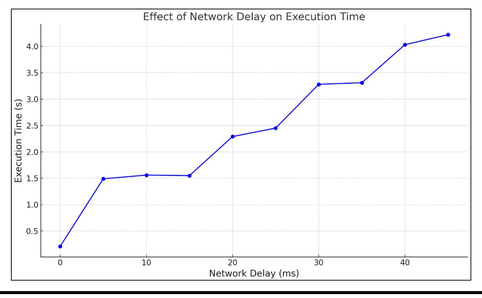
Objective

Simulate a busy server to evaluate strategies client devices can use for optimal results.

- Train & test on CIFAR-10 dataset
  - 60,000 images of airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks.
- Trained ResNet-18 model, split into 6 client and 12 server layers.

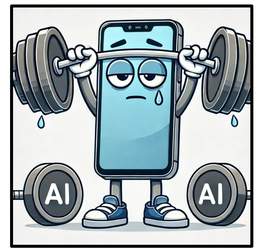


Effect of Network Delay



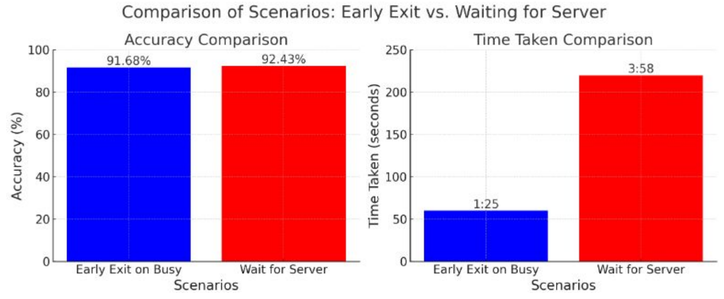
Problem

Running **complex AI** models on your phone can lead to overheating, battery draining quickly, and slower mobile processing. When should phones **offload to the cloud** and ask for help?



Results

- Shows our model is not overfitting to the training data
- Both early exit loss and validation loss decrease, indicating effective learning



- **Waiting for server** only increases accuracy from 91.68% to 92.43
- **Early exit prediction** maintains accuracy while reducing delay by 280%
- Early exit prediction **maintains accuracy** while **signficantly decreasing latency**

Main Takeaways

Effectiveness of Early Exiting when Server is Busy:

- Early exiting handles busy servers effectively with minimal drops in accuracy.
- Achieved 91.68% accuracy with reduced latency.

Application Suitability:

- Early exiting is better for real time applications: priortizing speed over accuracy
- For applications where accuracy is paramount, waiting for server resources is preferable

Scalability of Distributed Systems:

- The experiment highlights how balancing local processing (early exits) and centralized processing (server) can handle multiple clients efficiently.

Impact of Network Delay:

- As network delay increases, execution time also significantly increases linearly, indicating the importance of minimizing delay for real-time applications.

Experiment Setup

<u>Three Simultaneous Devices:</u>

- Run smaller ResNet18 model (first 6 layers)
- All attempt to offload to server

<u>One Server Node:</u>

- Runs remaining ResNet18 layers (last 12 layers)
- Handles offloaded requests

<u>Two Scenarios for Busy Server</u>

- Client **waits** for server to be available.
- Client **uses its own** prediction.

Future Work

- Compress models for mobile devices
- Explore dynamic thresholds
- Train model with diverse datasets for greater applicability and robustness
- Investigate RNN models
- Load balance offload requests

Special Thanks to Aliasghar Mohammadsalehi

National Science Foundation under grant CNS-2148104 and is supported in part by funds from federal agency and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program.

WINLAB