Distributed Deep Neural Network (DNN) Deployment

- DNN needs computing resources to be executed
- AI Application use Cloud to provide compactional resources
  - But not latency-sensitive
  - Data privacy issues
- Single-board computer execute part of the DNN in the network's edge.
Neural Service Function

- Create layers as functions
- Create sub-models
- Convert DNN to Directed Acyclic Graph (DAG)
- Topological sort
- Deploy layers
  - DNN layer Traversal (DAG Traversal)
  - Placement decision
  - Deploy a Neural layer as a function
DNN Deployment

Split DNN into layers

Layers Placement
Split DNN into layers

Pre-trained model → Model Compression → Split sub-models → Delivery

TensorFlow → TensorFlow Lite

Layer_1, Layer_2, Layer_3, Layer_4
Layers Placement

• Use dynamic programming to traverse over the DAG
• Greedy Nominator Heuristic (GNH)
  • Decide redundant deployment.
  • Uses Parsl to speed up the decision-making process
• Parsl
  • deploy the replicated layers in the Edge-Cloud resources
  • TensorFlow Lite inference with in Raspberry Pi 4 Model B (RPi4)
Experiment

- Deploy redundant Neural layers
  - speed up AI inference by up to 20%.
- Benchmark 16 DNN
  - with Parsl & TensorFlow Lite
  - Raspberry Pi 4 Model B (RPi4)
Thank You

AlmrushedO@cardiff.ac.uk