Edge Computing and Intelligence

Xiang Su (xiang.su@ntnu.no)
09.05.2022

Background

- AI applications are everywhere in our life
- Existing intelligent applications are computation-intensive, often based on cloud computing
- Large volumes of data is distributed at the edge of the network
- Privacy issues are more relevant, e.g., GDPR
- Edge computing emerges as an extension of cloud computing, providing computing service at edge
Edge Computing

- A new paradigm for distributed systems.
- “Edge computing refers to the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services.”

- Edge intelligence refers to a set of connected systems and devices for data collection, caching, processing, and analysis in proximity to where data is collected, with the purpose of enhancing the quality and speed of data processing and to protect the privacy and security of data.

- “The Internet of Things is a network of interconnected sensing and actuating devices that collect and share information using standard communication protocols.”
• **Edge intelligence:**
  - edge computing with AI capabilities.
  - Pushes the computing applications, data, and services away from centralized server to the network edge
  - Enables analytics and knowledge generation to occur close to the data sources
The benefits of edge intelligence result from its proximity to data sources and end users:

- Quick and efficient decision making by placing ML on edge nodes.
- Decisions according to local identity management and access control policies.
- Low and predictable latency for end users and applications;
- Secure and privacy-preserving services and applications;
- Long battery life and low bandwidth cost; and
- Scalability.
Edge Intelligence (vision)

- Edge Intelligence allows bringing data (pre-)processing and decision-making closer to the data source, which reduces delays in communication.
Edge Intelligence for Internet of Things

- By 2022, there will be 14.6 billion connected devices in the IoT (51% of connected devices)
  - Economic impact of $263 billion
- The data created by IoT devices will reach 283.5 exabytes (10^18 bytes) per year by 2022
  - 6% of global IP traffic
- Amount of data created (and not necessarily stored) by any device will reach 847 ZB per year by 2021
  - 218 ZB per year in 2016
  - Data created is >100X higher than data stored
IoT Value Chain (ecosystem)

- Hardware: Modules and objects - 25%
- Network: Equipment, cloud, OSS/BSS - 20%
- Connectivity - 5%
- Platforms - 15%
- Professional services and applications: Solutions, integration, operations and analytics - 35%

Key ecosystem players:
IoT Analytics

• Use data in real time to make our environments smarter
  • Sensors in vehicles and roads can monitor road and traffic conditions
  • Smart appliances and smart meters will enable fine-grained power usage monitoring and demand response
  • Cell phone cameras and microphones can measure environment (this information can be used to direct municipal resources)

• Need for big data analytics for IoT data
Data Center - Big Data Analytics

• Current State of the Art
  • Transfer data to the cloud and perform data analytics using a big data computing framework

• Challenges
  • How to parallelize computation across nodes?
  • How to distribute the data across nodes?
  • How to handle failures?
  • How to handle message loss and partitioning?
Data Center - Big Data Analytics

• A major reason for the success of cloud computing is the proliferation of data analytics frameworks

• Instead of writing code to address same challenges in every application:
  • Create abstraction that can express many types of computations
  • Develop a framework that implements this abstraction and hides messy details from application developers

• Example: Apache Spark, MapReduce (abstraction + framework)
Cloud Computing for the IoT

• Current approaches will not support the envisioned applications for the IoT

• Why not?
  • Too much data, not enough bandwidth
  • Latency and jitter are problematic for real-time sensing and control (IoT data is geographically distributed)
Edge Intelligence for IoT

Edge computing for Internet of Things:
Use Case Taxonomy

Edge Architectures

- Resource-rich servers deployed close to the end-devices
- Resources from heterogeneous nodes at the edge, including the end-devices themselves
- Federation of resources at the edge and centralized data centers.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Edge nodes</th>
<th>Edge network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloudlets [2]</td>
<td>Compact-size data centers deployed on WiFi access points, femtocells or LTE base stations</td>
<td>WiFi, 3G or LTE</td>
</tr>
<tr>
<td>Mobile cloudlets [13]</td>
<td>Compact-size data centers on cars</td>
<td>3G or LTE</td>
</tr>
<tr>
<td>Multi-access edge computing [19]</td>
<td>Servers deployed in the radio access network</td>
<td>3G, LTE, WiFi or other access technologies</td>
</tr>
<tr>
<td>Fog computing [3]</td>
<td>Heterogeneous nodes including high-end servers, routers, access points and set-top boxes</td>
<td>Multiple wireless access technologies including WiFi, 3G and LTE</td>
</tr>
<tr>
<td>Mobile cloud [20]</td>
<td>Neighboring mobile nodes form a cloud with one device chosen as resource coordinator</td>
<td>Local networking through WiFi or Bluetooth; Internet connectivity with WiFi, 3G and LTE</td>
</tr>
<tr>
<td>Edge cloud [21]</td>
<td>Compute or storage nodes deployed in the edge network and federated to cloud data centers</td>
<td>Home/enterprise networks and WiFi hotspots</td>
</tr>
<tr>
<td>FUSION [10]</td>
<td>Service nodes deployed on access points, local data centers and centralized data centers</td>
<td>Not defined</td>
</tr>
</tbody>
</table>

## Mobile Edge Architectures

<table>
<thead>
<tr>
<th>MEC concept</th>
<th>Control entity</th>
<th>Control manner</th>
<th>Control placement</th>
<th>Computation/storage placement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCC</td>
<td>SCM</td>
<td>Centralized, decentralized hierarchical (depending on SCM type and placement)</td>
<td>In RAN (e.g., at eNB) or in CN (e.g., SCM collocated with MME)</td>
<td>SCeNB, cluster of SCeNBs</td>
</tr>
<tr>
<td>MMC</td>
<td>-</td>
<td>Decentralized</td>
<td>MMC (eNB)</td>
<td>eNB</td>
</tr>
<tr>
<td>MobiScud</td>
<td>MC</td>
<td>Decentralized</td>
<td>Between RAN and CN</td>
<td>Distributed cloud within RAN or close to RAN</td>
</tr>
<tr>
<td>FMC</td>
<td>FMCC</td>
<td>Centralized, decentralized hierarchical (option with hierarchical FMCC), decentralized (option without FMCC controller)</td>
<td>Collocated with existing node (e.g., node in CN) or run as software on DC</td>
<td>DC close or collocating with distributed CN</td>
</tr>
<tr>
<td>CONCERT</td>
<td>Conductor</td>
<td>Centralized, decentralized hierarchical</td>
<td>N/A (it could be done in the same manner as in FMC concept)</td>
<td>eNB (RIE), regional and central servers</td>
</tr>
<tr>
<td>ETSI MEC</td>
<td>Mobile edge orchestrator</td>
<td>Centralized</td>
<td>N/A (the most feasible option is to place control into CN)</td>
<td>eNB, aggregation point, edge of CN</td>
</tr>
</tbody>
</table>

ESTI MEC Reference Architecture

- Functional elements (VMs on datacenter (NFV)) and reference points
- Mobile edge orchestrator (system level management of available computing/storage/network resources)
- Mobile edge platform manager (server level management of application lifecycle, application rules, and service authorization)
- Virtualization infrastructure manager allocation (server level management and release of the virtualized computation/storage resources)
- MEC Server placement (ENB, Cell aggregation site, edge of core network)

Practical Implementation

- Nokia Airframe open edge server (for BS)
- 130.6 x 440 x 430 mm
- 5 server sled slot

- 1U Server
  - 24 cores/2.4GHz
  - 8 DIMM Slots
  - 2 hot-plug SATA Slots
  - Support Nvidea Tesla T4 GPU
## Practical Costs

### Table 11-14: Edge Cloud CAPEX costs

<table>
<thead>
<tr>
<th>Metric</th>
<th>Unit</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed costs for initial set up of an installation at edge cloud site</td>
<td>10,100</td>
<td>£ Based on estimates for base station hotel set-up costs, and LLU costs for metering and site visit. Costs include: power supply distribution boards, sockets, lighting, enclosure, overhead racking and cabling.</td>
</tr>
<tr>
<td>Fixed costs to set up a cabinet/rack at the edge cloud site</td>
<td>21,400</td>
<td>£ Includes power distribution, Air Conditioning set-up, space set-up, AC distribution and cabinet.</td>
</tr>
<tr>
<td>Fixed costs per server</td>
<td>6,500</td>
<td>£ Maximum of 16 servers per cabinet. Assumes 35% discount. This is equivalent to just under £490/installed core (for a fully equipped cabinet)</td>
</tr>
</tbody>
</table>

### Table 11-15: Edge Cloud OPEX costs

<table>
<thead>
<tr>
<th>Metric</th>
<th>Unit</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Licensing and maintenance</td>
<td>10%</td>
<td># Assumed licensing and maintenance of 10% of the capex of the active equipment capex needed to process the sites feeding into each edge cloud.</td>
</tr>
<tr>
<td>Transport</td>
<td>1150</td>
<td>£ This is assumed dark fibre costs – this could be an underestimate and would depend upon the capability to support all the fibres feeding into the site. Fibre communications modules are installed in each server.</td>
</tr>
<tr>
<td>Standing charges / edge cloud</td>
<td>6,300</td>
<td>£ Security and working practices audit (assumed annual per site), and one site maintenance and update visit (per site))</td>
</tr>
</tbody>
</table>
| Site rent and utilities/cabinet       | 6,600 | £ This includes, rental and service charge (for a cabinet space, and working space), Standby Power / cabinet, Power connection (rental), Electricity use (of a fully stacked cabinet). Electricity use is the dominant cost.
Edge Landscape

- https://www.stateoftheedge.com/projects/landscape/

- Traffic, Platform, Infrastructure, Hardware, Connectivity, Real Estate, etc.
Edge intelligence in smart cities
Edge intelligence for Scalable Air Quality Monitoring
Distributed ML on Edge

- Natural use case for edge intelligence
- Federated learning
  - Training on local data
  - Send model updates
  - Train/Val/Test on subsets of clients
  - Data privacy
  - Heterogenous devices w/ constraints
  - Communication overhead
- Implementations
  - Fed Avg (averaging local SGD updates)
  - TensorFlow Federated
  - PySyft
  - LEAF Benchmark

Figure 1: Cloud-based, edge-based and client-edge-cloud hierarchical FL. The process of the FAVG algorithm is also illustrated.
## Communication Overhead and Resource Allocation in FL

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Ref.</th>
<th>Key Ideas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge and End Computation</td>
<td>[23]</td>
<td>More local updates before communication</td>
</tr>
<tr>
<td></td>
<td>[97]</td>
<td>Reference to global model for faster convergence</td>
</tr>
<tr>
<td></td>
<td>[98]</td>
<td>Intermediate edge aggregation before FL server aggregation</td>
</tr>
<tr>
<td>Model Compression</td>
<td>[88]</td>
<td>Structured and sketched updates for participant-to-server communication</td>
</tr>
<tr>
<td></td>
<td>[93]</td>
<td>Lossy compression and federated dropout for server-to-participant communication</td>
</tr>
<tr>
<td>Importance-based Updating</td>
<td>[94]</td>
<td>eSGD to selectively communicate parameters that reduce training loss</td>
</tr>
<tr>
<td></td>
<td>[90]</td>
<td>CMFL to selectively communicate parameters based on signs of parameters compared to global parameters</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Issue</th>
<th>Ref.</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participant Selection</td>
<td>[78]</td>
<td>FedCS to select participants based on computation capabilities</td>
</tr>
<tr>
<td></td>
<td>[114]</td>
<td>Hybrid-FL to select participants for IID data collection</td>
</tr>
<tr>
<td></td>
<td>[115]</td>
<td>DRL to determine resource consumption by participants</td>
</tr>
<tr>
<td></td>
<td>[119]</td>
<td>Fair resource allocation</td>
</tr>
<tr>
<td></td>
<td>[121], [123]</td>
<td>Participant selection based on distance threshold to increase SNR in BAA</td>
</tr>
<tr>
<td></td>
<td>[124]</td>
<td>Participant selection to keep signal distortion low</td>
</tr>
<tr>
<td>Adaptive Aggregation</td>
<td>[129], [111]</td>
<td>Asynchronous FL where model aggregation occurs once local updates are received by FL server</td>
</tr>
<tr>
<td></td>
<td>[65]</td>
<td>Adaptive global aggregation frequency</td>
</tr>
<tr>
<td>Incentive Mechanism</td>
<td>[130]</td>
<td>Stackelberg game and relay network to support model update transfer</td>
</tr>
<tr>
<td></td>
<td>[132]</td>
<td>Stackelberg game to incentivize computation resource usage in FL training</td>
</tr>
<tr>
<td></td>
<td>[133], [62]</td>
<td>Contract theory, reputation mechanisms and blockchain</td>
</tr>
</tbody>
</table>
Use Case: Android GBoard

- Learning next-word prediction for Android keyboard
- Local keyboard cache vs. server log data (truncated, only google apps)
- FedAvg (averaging local SGD updates)

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1 recall [%]</th>
<th>Top-3 recall [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-gram</td>
<td>5.24 ± 0.02</td>
<td>11.05 ± 0.03</td>
</tr>
<tr>
<td>Server CIFG</td>
<td>5.76 ± 0.03</td>
<td>13.63 ± 0.04</td>
</tr>
<tr>
<td>Federated CIFG</td>
<td>5.82 ± 0.03</td>
<td>13.75 ± 0.03</td>
</tr>
</tbody>
</table>

Table 5. Prediction impression recall for the server and federated CIFG models compared with the n-gram baseline, evaluated in experiments on live user traffic.

When to use Federated Learning?

1. The task labels don’t require human labelers but are naturally derived from user/machine interaction
2. The training data is privacy sensitive
3. The training data is too large to be feasibly collected centrally

Traditional intelligence vs. Edge intelligence
Content

- Solo training
- Collaborative training
- Update frequency
- Update cost
- Privacy & security

- Training architecture
- Training acceleration
- Training optimization
- Uncertainty estimates
- Applications

- Offloading strategy
- Model design
- Model compression
- Inference acceleration
- Applications

- Edge offloading
- D2C offloading
- D2E offloading
- D2D offloading
- Hybrid offloading

- Offloading strategy
- Inference strategy

- Cache replacement
- Cache deployment
- Cache content

- Data
- Computation

- Cache content
- Caching at macro BS
- Caching at micro BS
- Caching at devices

- Architecture search
- Human-invented architecture
- Low-rank approximation
- Knowledge distillation
- Compact layer design
- Network pruning
- Parameter quantization
- Hardware acceleration
- Software acceleration

- Update cost
- Privacy & security

- Collaborative training
- Solo training

- Training architecture
- Training optimization

- Edge Intelligence

- Edge
- Inference

- Training acceleration
- Training optimization

- D2C offloading
- D2E offloading
- D2D offloading
- Hybrid offloading
Data generated by mobile users and collected from surrounding environments is collected and stored at the edge of the network. Such data is processed and analysed by intelligent algorithms to provide services for end users.
Edge Caching—data collection
The model/algorithm is trained either on a single device (solo training), or by the collaboration of edge devices (collaborative training) with training sets cached at the edge.

- Acceleration module speeds up the training
- Optimization module solves problems in training (update frequency, update cost, and privacy and security issues)
- Uncertainty estimates module controls the uncertainty in training.
Edge Training—architecture

Master-Slave architecture
Edge Training—acceleration

hourglass accelerator model
Edge Inference

- AI models/algorithms are designed either by machines or humans.
- Models could be further compressed through compression technologies: low-rank approximation, network pruning, compact layer design, parameter quantization, and knowledge distillation.
- Hardware and software solutions are used to accelerate the inference with input data.
Architecture—Network pruning

- Prune to delete unimportant parameters, since not all parameters are equally important in highly precise deep neural networks.
- Consequently, connections with less weights are removed, which converts a dense network into a sparse one.
Architecture—Low rank approximation

- Use the multiplication of low-rank convolutional kernels to replace kernels of high dimension
- Matrix can be decomposed into the multiplication of multiple matrices of smaller size
Architecture—Compact layer design

- Design compact layers in neural networks
- Effectively reduce the consumption of resources (memory and computation)
High precision parameters in neural networks are not always necessary to achieve high performance

Especially when these highly precise parameters are redundant
Knowledge distillation is based on transfer learning.

- Trains a neural network of smaller size with the distilled knowledge from a larger model.
- The large and complex model is called the teacher model while the compact model is referred to as the student model.
- Transferring knowledge from the teacher network to the student network.
• Parallelising inference tasks to available computing resources, such as CPU or GPU
Edge Offloading

- Edge offloading part of bottom layer of edge intelligence
- Provides computing services for edge caching, edge training, and edge inference
- Potential architectures include D2C, D2E, D2D, and hybrid computing
Architecture—Edge Offloading

- **D2C**: Direct to Cloud
- **D2E**: Direct to Edge
- **D2D**: Device to Device
- **Hybrid**: A combination of D2D and D2E services

- Cloud server
- Edge server
- Edge device
- Network layers
- Partial model
- Network layer
Development

Publication volume over time. These curves show the trend of publication volume in edge caching, edge training, edge computing, edge inference, and edge intelligence, respectively.
## Edge Intelligence Challenges

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Description</th>
<th>Potential Research Directions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Scarcity at Edge</td>
<td>Data collected at edge is often unlabeled, poor quality, or both</td>
<td>Shallow model with small dataset</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Incremental learning or transfer learning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data augmentation-based methods</td>
</tr>
<tr>
<td>Data Consistency at Edge</td>
<td>Data collected at edge may be inconsistent due to different sensing environments or sensor heterogeneity</td>
<td>Data augmentation-based methods</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Representation learning</td>
</tr>
<tr>
<td>Adaptability of Staticly Trained Models</td>
<td>Models trained on a central server or trained only on edge do not adapt well to new unknown data or larger areas</td>
<td>Lifelong ML</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Knowledge sharing between models</td>
</tr>
<tr>
<td>Security and privacy</td>
<td>Data sent to unknown edge devices for processing thus potential for sensitive data privacy leakage. Computing tasks offloaded to edge could contain malicious code.</td>
<td>Authentication systems including credit-based</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Federated learning and differential privacy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Homomorphic encryption (for AI on encrypted data)</td>
</tr>
<tr>
<td>Incentive mechanism</td>
<td>Need to incentivize the edge devices to share the data and intelligence through socioeconomic means, and thus make the ecosystem healthy</td>
<td>Sharing mechanism by virtual resource pool and multi-agent distributed learning approach</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lightweight blockchain consensus protocol to smartly record the price and disseminate the revenue</td>
</tr>
<tr>
<td>Edge Intelligence in 6G</td>
<td>Need to consider the integration of edge intelligence into the future 6G networks</td>
<td>Edge intelligence placement in the context of global coverage of 6G (satellite networks with high bandwidth, etc.)</td>
</tr>
</tbody>
</table>