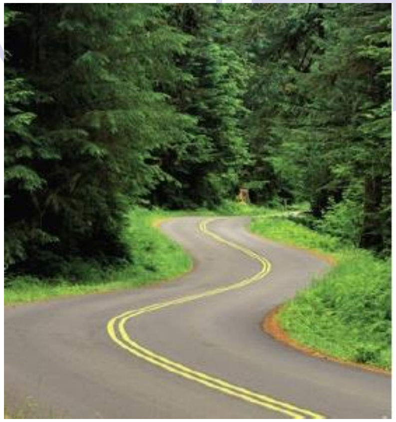
On Optimal Partitioning and Scheduling of DNNs in Mobile Edge/Cloud Offloading

Jie Wu
Dept. of Computer and Information Sciences
Temple University, USA

IEEE TCCN

Roadmap



1. On Problem Solving
2. Edge/Cloud Computing + AI
3. Optimal Scheduling
4. Optimal Partition and Scheduling
5. Conclusions and Future Work

1. On Problem Solving

How to Solve It (Poyla, 1945)

If you can't solve a problem, then there is an easier problem you can solve: find it.

Is Computing An Experimental Science ? (Milner, 1986)

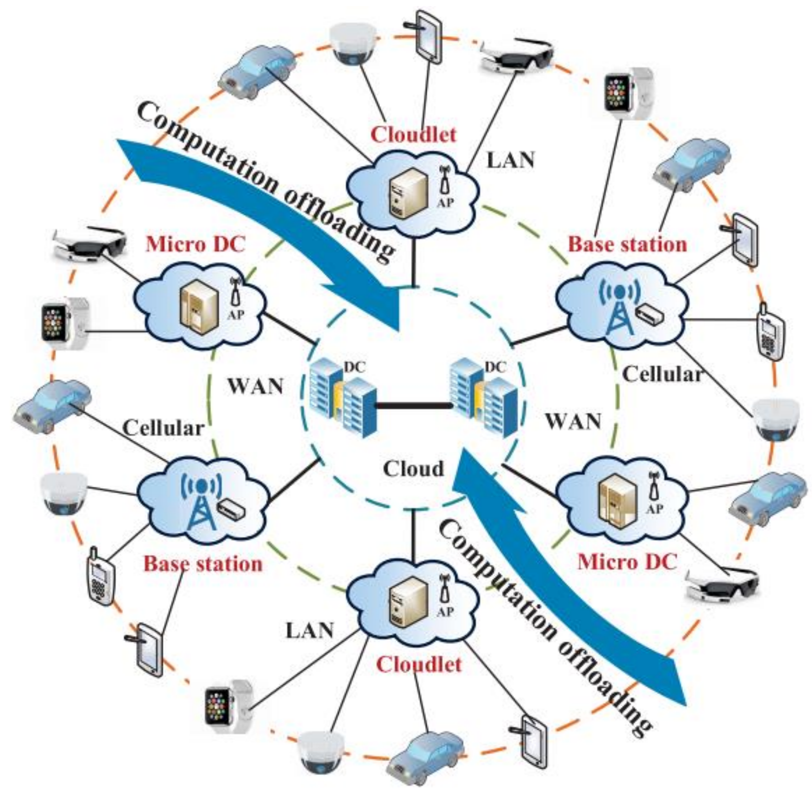
A theory can only emerge through protracted exposure to application.

Ideas and applications developed side-by-side

Edge-computing + ML algorithms: traditional solutions

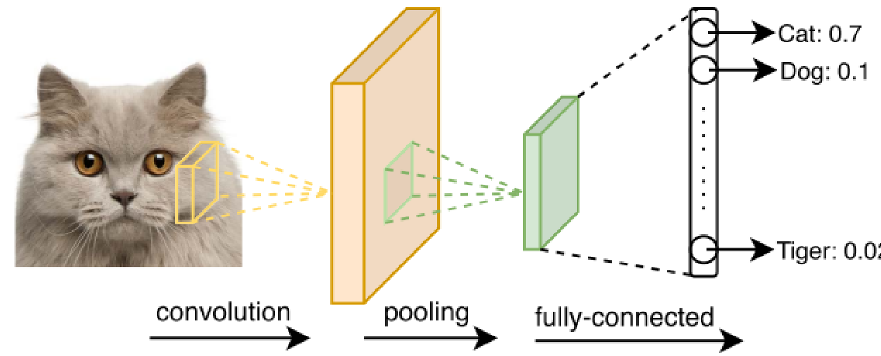
2. Edge/Cloud Computing + AI

- Edge/Cloud Computing
  - Application-driven: AR/VR, video analytics using IoTs
  - Better QoE: mobile/edge device
  - Key indicators: latency, accuracy, energy, and privacy
  - Latency-sensitive
    - How to bring rich computation resources to mobile users?
    - How IoTs contribute to the ML training and inference?

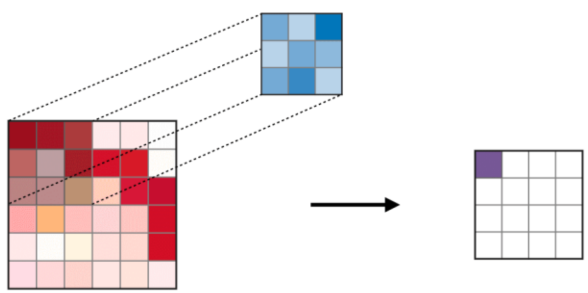


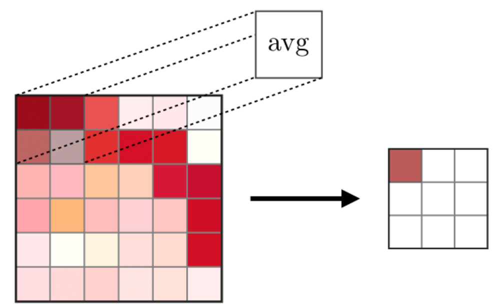
50 billion IoTs: connected intelligence

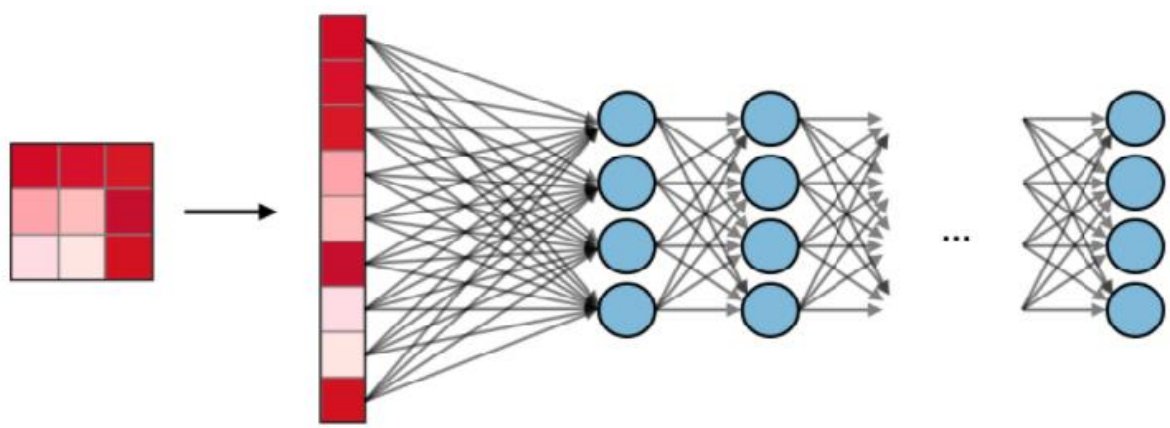
Convolution NNs



- CNNs (image classification)
- convolution (filtering), pooling (max/avg), fully-connected (neurons)

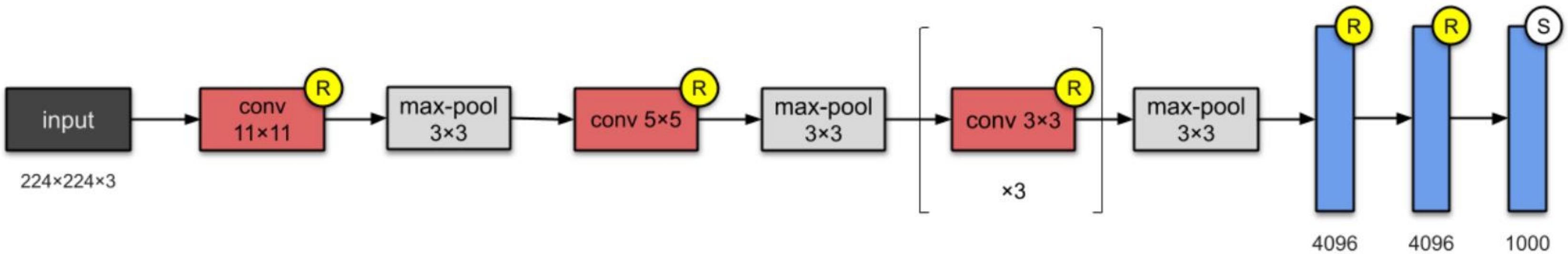




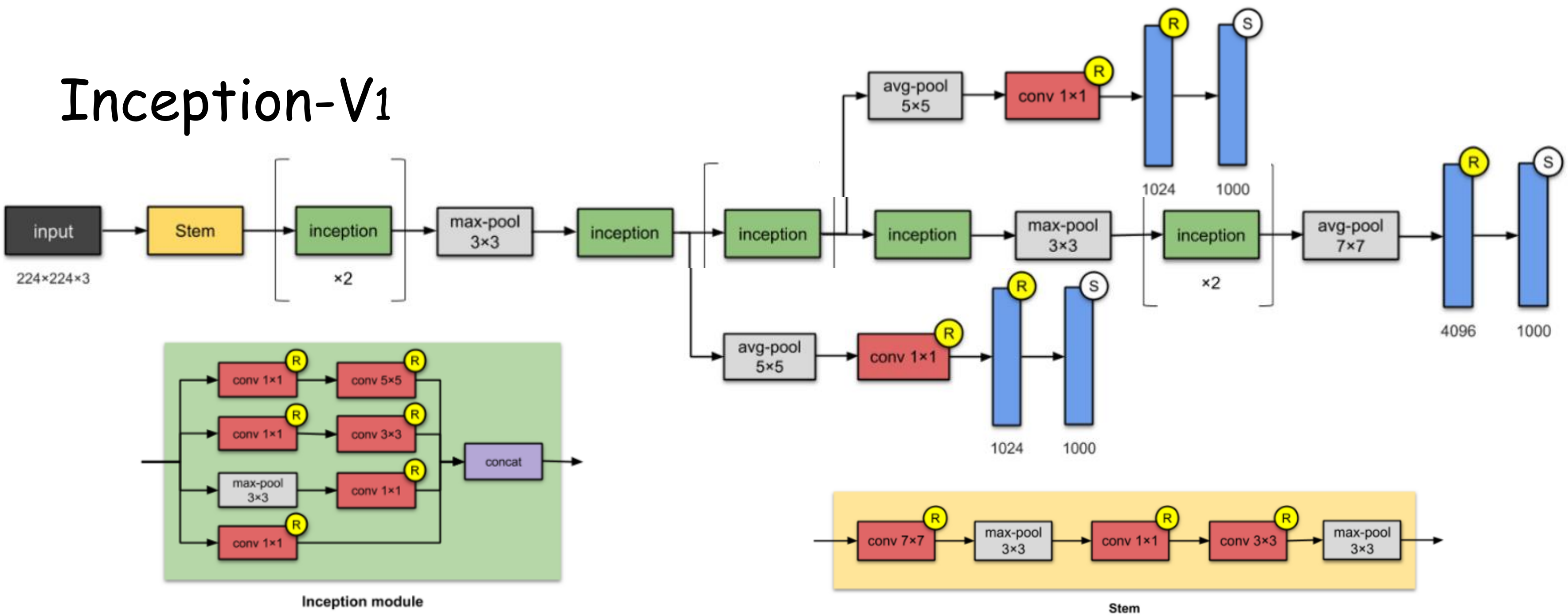


Sample CNNs

AlexNet (Red: CONV, Gray: POOL, Blue: FC)



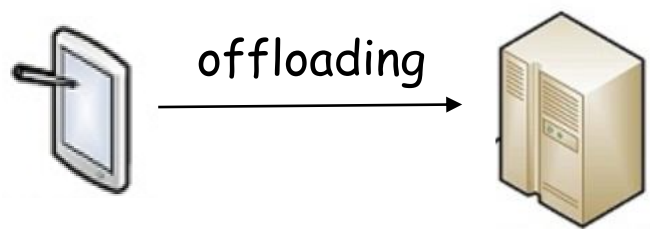
Inception-V1



Offloading

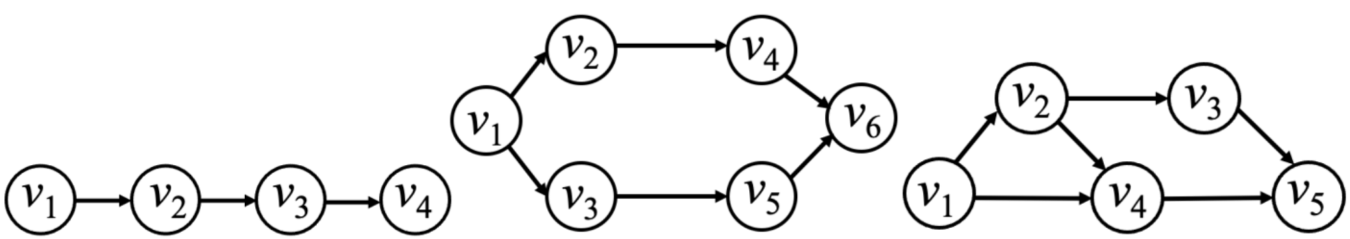
- Three-stage collaborative computation offloading
  - Local computation: processing on local devices
  - Communication: transmitting intermediate DNN layers' outputs
  - Remote computation: completing the remote processing in cloud

- Three models
  - On-device optimization
  - Cloud-only offloading
  - Mixed-mode offloading



DNN Inferencing

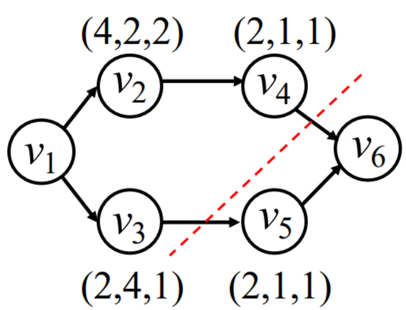
- Deep Neural Networks (DNNs)
  - Technologies: GPU (graphic) and TPU (tensor)
- AI applications
  - Computer vision: AlexNet, VGG-16, Inception, GoogLeNet
    Siamese, Multi-Stream, and RandWire
  - Natural language processing: ChatGPT, GPT-4
- Graph models of DNNs



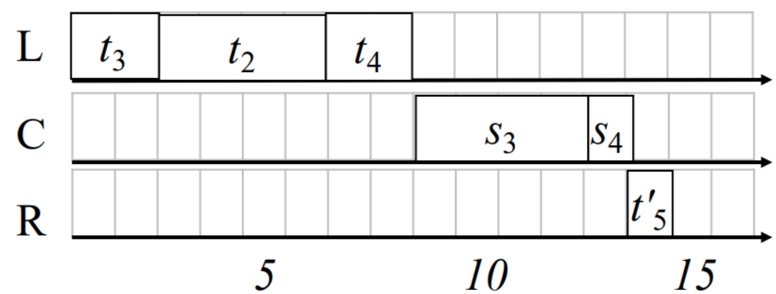
(a) line (b) multi-path (c) DAG

Offloading Samples

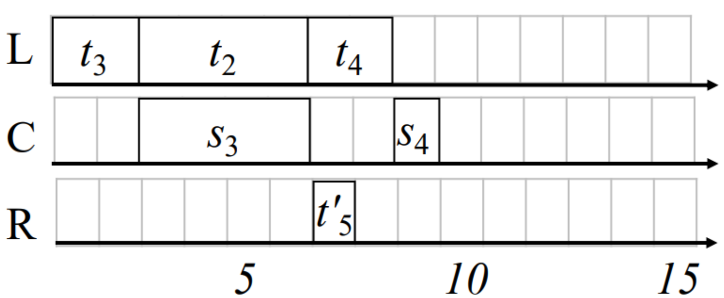
- Given a partition (i.e., cut)
  - Coarse-grained pipeline: local, communication, and remote
  - Fine-grained pipeline: path-based (rather than phase-based)

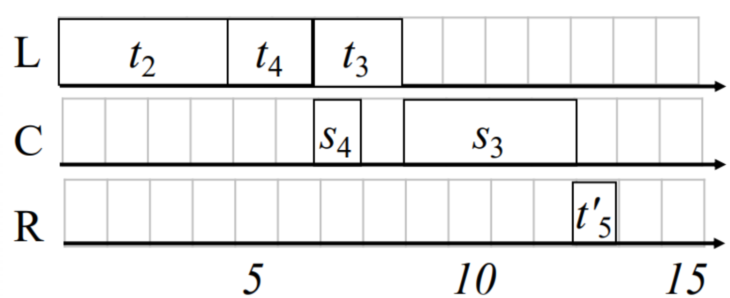


(a) a DNN



(b) no fine-grained pipeline





3. Optimal Scheduling

- DNN Computation Offloading Optimization (DCOO)
  - DCOO: minimum makespan for a given partition (i.e., cut)

- Cases of DNN
  - Line-structure: trivial
  - Multi-path: hard
  - DAG: hard

Theorem 1: DCOO is NP-hard for a multi-path DNN.

Proof: Reduce 3-machine flow-shop to DCOO.

Extended Johnson Algorithm (EJA)

Path p(i) in three stages

- $P_1(i)$, $P_2(i)$, $P_3(i)$

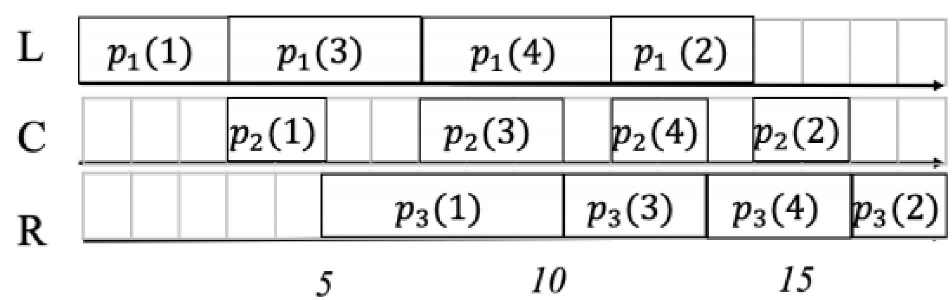
Linear solution (EJA)

- Dividing paths into H and L
- E.g., H = {1}, L = {3, 4, 2}

Algorithm 1 Extended Johnson Algorithm (EJA)

```
H ← L ← φ
for i = 1 to m do
    if p1(i) + p2(i) ≤ p2(i) + p3(i) then
        H = H ∪ p(i)
    else
        L = L ∪ p(i)
Sort H increasingly based on p1(i) + p2(i)
Sort L decreasingly based on p2(i) + p3(i)
Concatenate H and L to obtain σ
```

Path	$p_1(i)$	$p_2(i)$	$p_3(i)$
$i = 1$	3	2	5
$i = 2$	3	2	2
$i = 3$	4	3	3
$i = 4$	4	2	3



Optimality

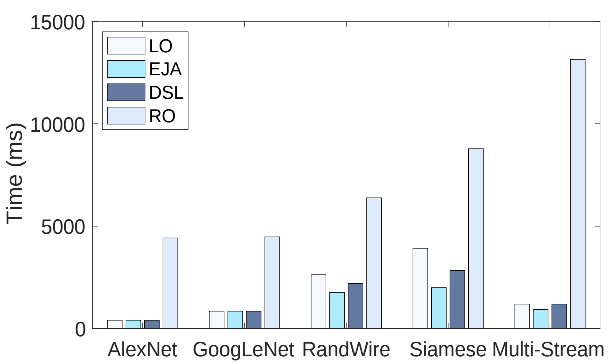
Theorem 2*: If stage 2 is dominated by either stage 1 or 3, $\max\{\min p_1(i), \min p_3(i)\} \geq \max p_2(i)$, EJA is optimal.

If Theorem 2 fails, EJA still achieves an approximation ratio of 5/3.

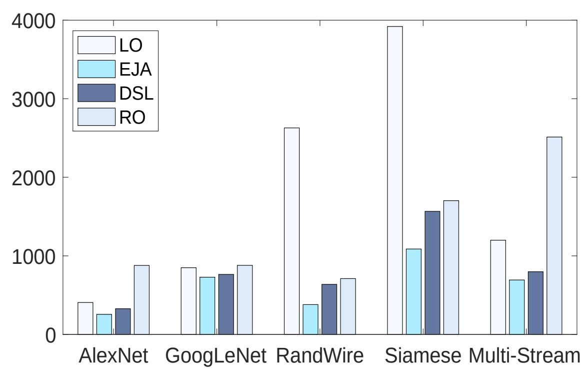
Path	$p_1(i)$	$p_2(i)$	$p_3(i)$
$i = 1$	3	2	5
$i = 2$	3	2	2
$i = 3$	4	3	3
$i = 4$	4	2	3

Simulation

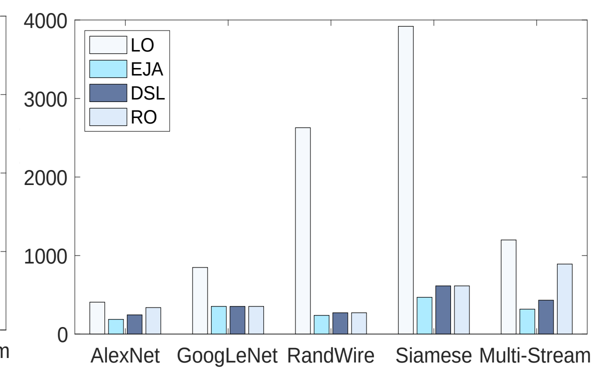
- Local and Cloud
  - Local: Raspberry Pi (and Nexus 4), Cloud: Amazon EC2
  - PyTorch: open-source ML framework
- Algorithms
  - LO: local only, EJA: Extended Johnson's Algorithm, DSL: coarse-grained pipeline, RO: remote only



3G (1.1 Mbps)



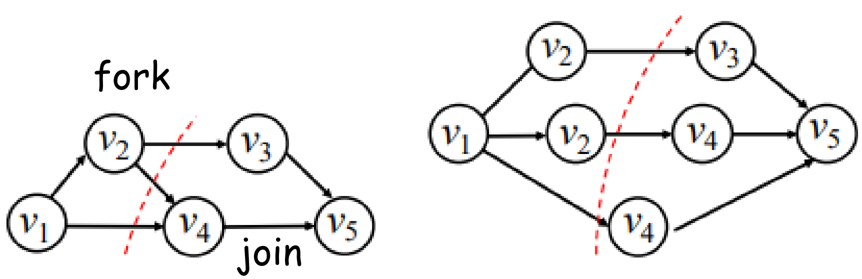
4G (5.85 Mbps)



WiFi (18.88 Mbps)

Extensions: DAG

- General structure: DAG
  - Conversion to multi-path
  - Replicated nodes at join and fork
- Heuristic solution
  - Scheduling: EJA on multi-path
  - Execution: Replicated node executed once (the first time)



(a) Before Conversion, G (b) After Conversion, G'

Multiple DNNs Offloading

Internet of Vehicles: smart city

- Autonomous driving systems: perception is a key
- Multiple cameras/sensors: multiple (identical) DNNs
- V2X: V (vehicle); X for I (infrastructure), N (network), or P (pedestrian)

4. Optimal Partition and Scheduling

- Multiple line-structure DNNs
  - AlexNet and VGG-16
  - Video analytics and AR/VR
- Optimal partition and scheduling
  - Brute force: $O(k^n)$

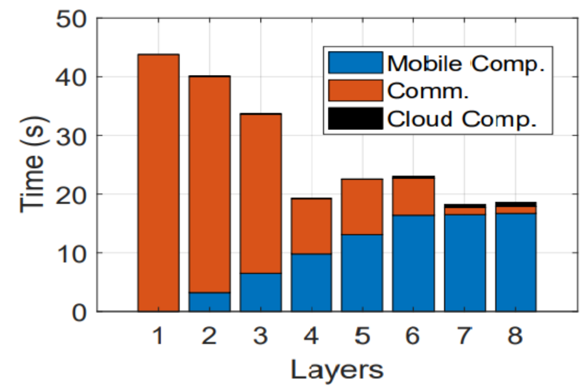
    n: # of copies, k: # of layers
- Existence of a better solution?
  - Exploring special application properties

Johnson Algorithm (JA)

- Closer look at the optimality for EJA
  - $\max\{\min p_1(i), \min p_3(i)\} \geq \max p_2(i)$
- However, $p_3(i) \approx 0$, reduced to 2-stage pipeline

Algorithm 2 Johnson Algorithm (JA)

$H \leftarrow L \leftarrow \phi$
for $i = 1$ to m **do**
  if $p_1(i) \leq p_2(i)$ **then**
    $H = H \cup p(i)$
  else
    $L = L \cup p(i)$
Sort H increasingly based on $p_1(i)$
Sort L decreasingly based on $p_2(i)$
Concatenate H and L to obtain σ

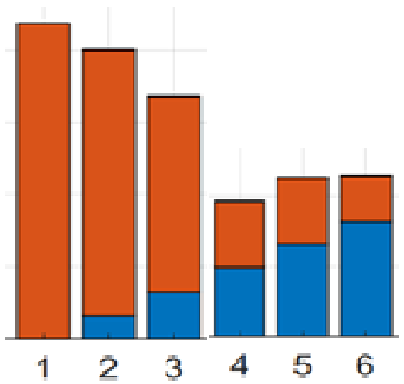


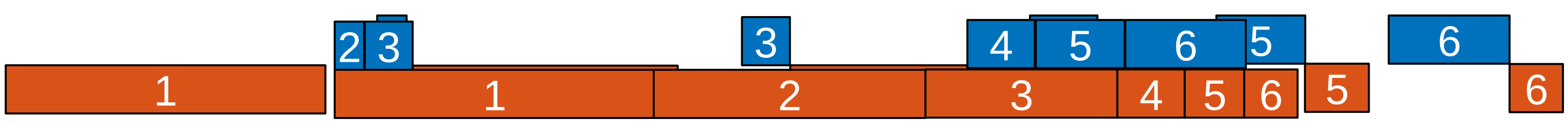
AlexNet

Johnson, Optimal Two- and Three-Stage Production Schedules With Set-up Time Included, *Naval Research Logistics Quarter, 1954.*

JA in Illustration

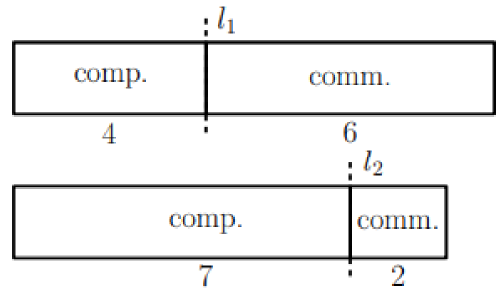
- Optimality is guaranteed: JA on 2-stage pipeline
- First six layers of AlexNet
  - One copy for each partition: 6 copies
  - H = {1, 2, 3}, increasing order of blue (H: comm.-dominate)
  - L = {4, 5, 6}, decreasing order of red (L: comp.-dominate)



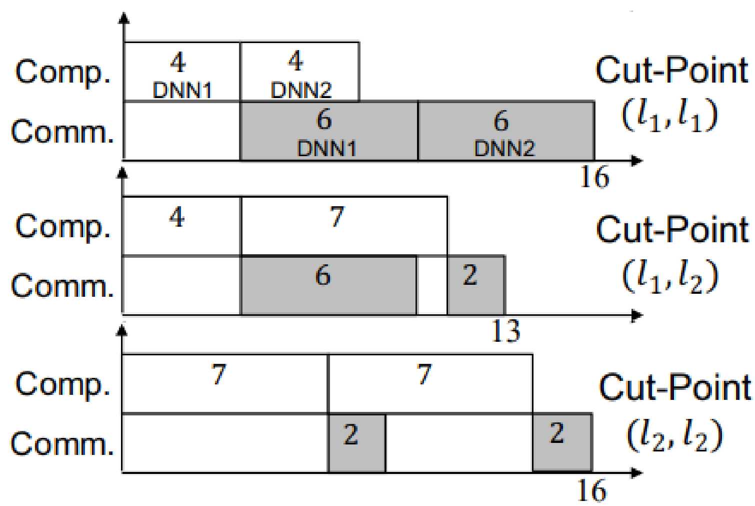


Multiple Line-Structure Example

- Two copies of line-structure DNN

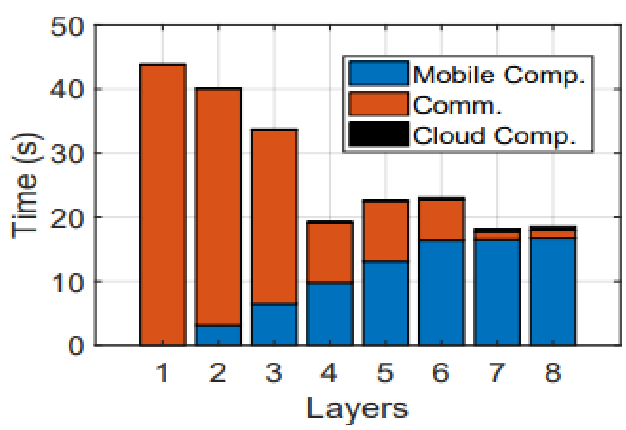


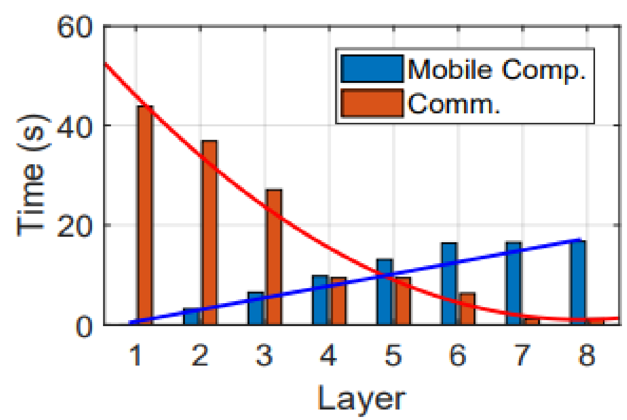
- Three possible partitions and scheduling
  - Gaps in the first and last pairs of comp. and comm.



Special Application Property

- Line-structure
  - Computation time: linear increasing (convex) function
  - Communication time: monotonic decreasing convex function
- Computation vs. communication
  - Data size: 2 – 12 MB
  - Speed (uplink): 2-5 Mpbs (4G) and 6-54 Mpbs (WiFi)





Optimization Approximation

- Two functions
  - Comp. and comm. are convex: one increasing, one decreasing

Theorem 3: A uniform partition of n line DNNs at the intersection will guarantee an approximation of $1 + \frac{1}{n}$.

Proof: convex optimization

- Intersection has the min {max {comp., comm.}}
- Strong duality, then KKT condition, the uniform partition at the intersection has the min max { ∑comp., ∑comm. }
- 1/n is caused by the gaps in the first and last pairs

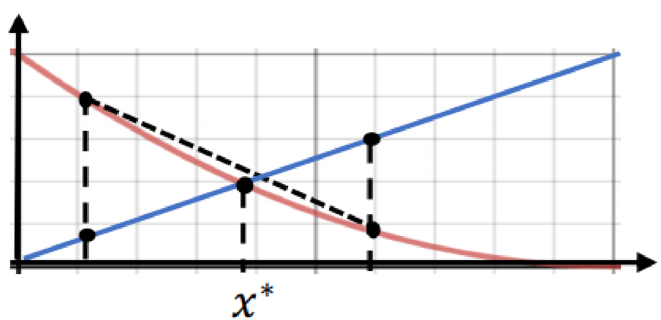
Duan and Wu, Joint Optimization of DNN Partition and Scheduling for Mobile Cloud Computing, *Proc. of ICPP*, 2021.

Optimization

- Informal proof
  - Pair-wise "merge" and replaced by the middle-point

$$\frac{f(x)+f(x')}{2} \geq f(\frac{x+x'}{2})$$

  - The height of the intersection ≤ any max {comp., comm.} of a partition

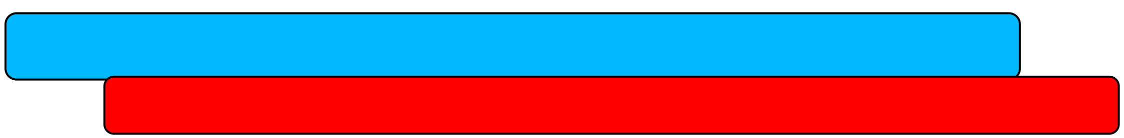


  - Two gaps, first pair in comm and last pair in comp.: when $n \rightarrow \infty, 1+1/n$ approaches 1

Insight

- Comp. (blue) and comm. (orange)
  - max {blue sum, orange sum}

  - However, there is a delay gap

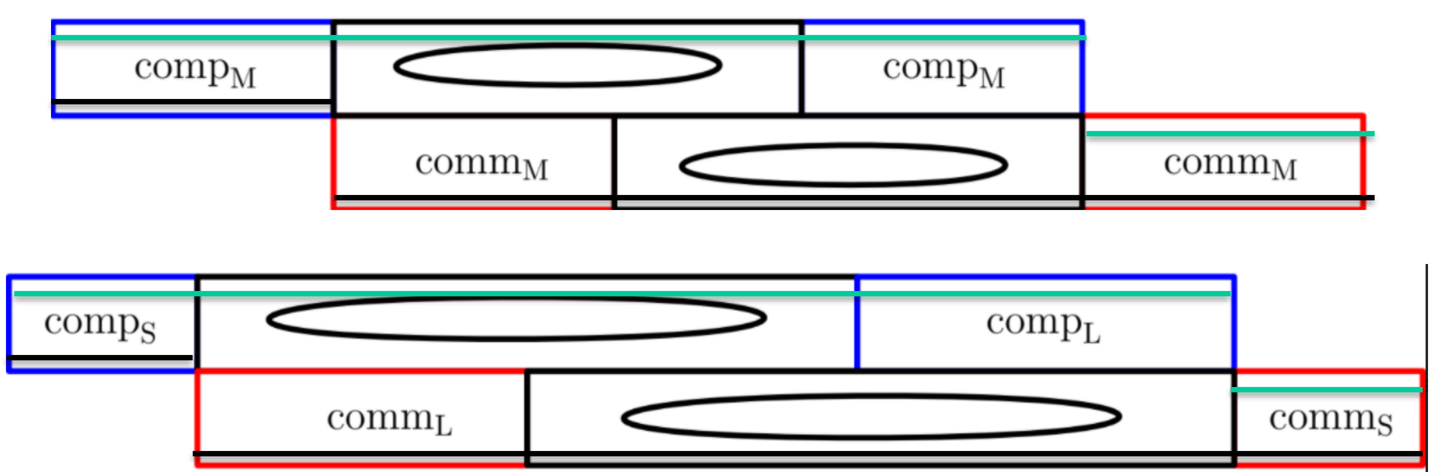


Sufficient Condition

- A set of given partitions
  - Left/right most partition: $comm_l$ and $comp_s$ / $comm_s$ and $comp_l$
- Intersection partition: $comm_m$ and $comp_m$

Theorem 4: The uniform partition beats the given set if

$3comp_m < comp_s + comp_l + comm_s$ and $3comm_m < comp_s + comm_l + comm_s$



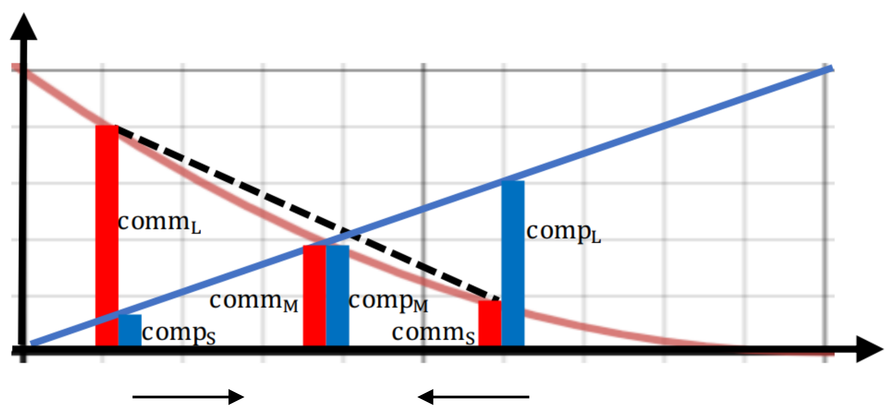
Extended Sufficient Condition

Theorem 5: The uniform partition beats the given set if

$(k+2)\ comp_m < k\text{-prefix}.comp_s + k\text{-postfix}.comp_l + k\text{-postfix}.comm_s$

$(k+2)\ comm_m < k\text{-prefix}.comp_s + k\text{-prefix}.comm_l + k\text{-postfix}.comm_s$

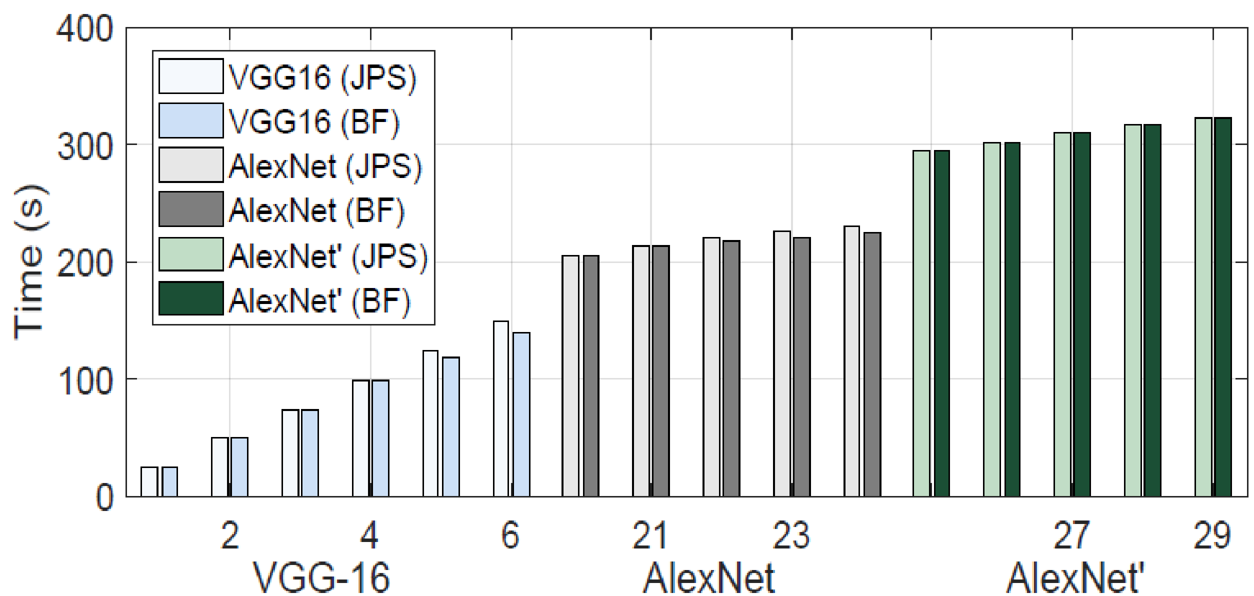
K-prefix/k-postfix: summation of k left/right most partition



Duan and Wu, Optimizing Job Offloading Schedule for Collaborative DNN inference, to appear in *IEEE Transactions on Mobile Computing*, 2023.

Simulation

- Partition methods
  - Joint Partition and Scheduling: JPS, Brute Force: BF
- Application
  - VGG-16, AlexNet, and AlexNet' (curve fitting) with n = 1, …, 29

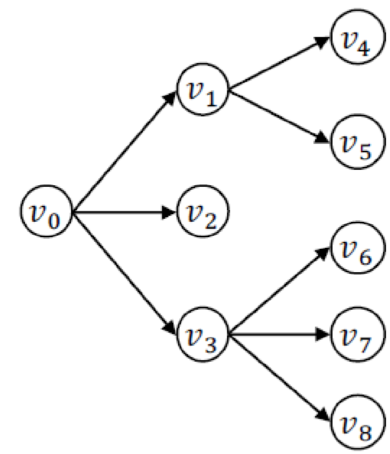


Extension: Tree-structure DNNs

- Merge schedules of subtrees bottom-up

  (cut at leaves, i.e., $p_3(i) = 0$)

  - Multiway merge of child lists
    - One node at a time, based on Johnson's rule
  - Aggregate computation
    - Root of a subtree and its first child



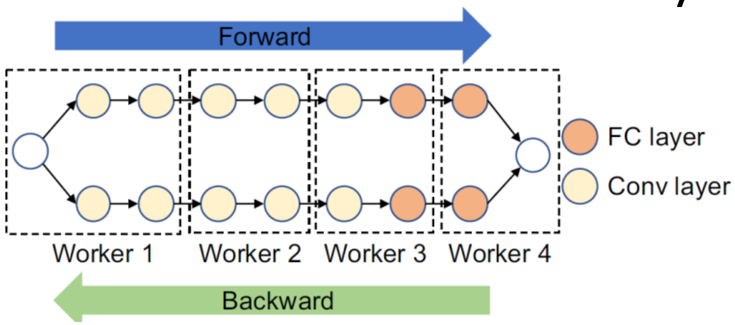
> Theorem 6: The schedule generated by the recursive merging approach is optimal for tree-structure DAGs.

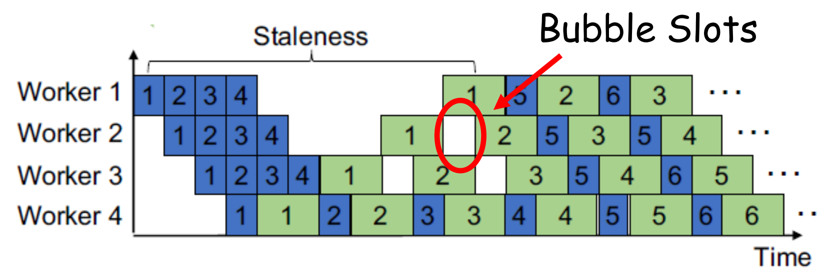
Duan and Wu, Computation Offloading Scheduling for Deep Neural Network Inference in Mobile Computing, *Proc. of ACM/IEEE IWQoS*, 2021.

Extension: Inference/Training

Inference forward pass/training backward pass

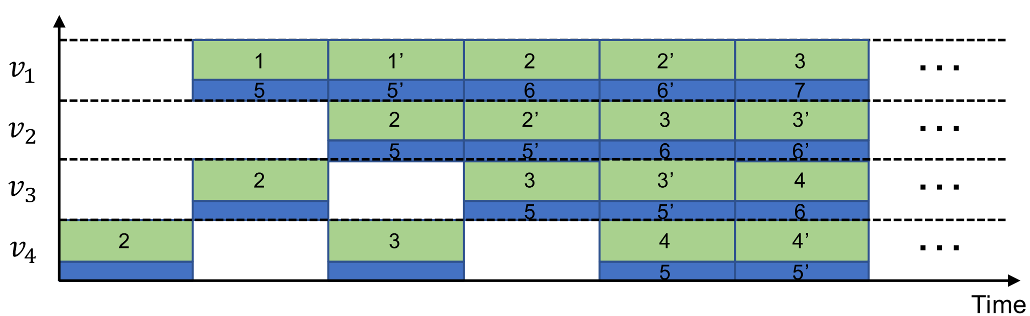
- Reduce resource idle time by adjusting the ratio of resources





Aligning Pipeline with Resource Allocation

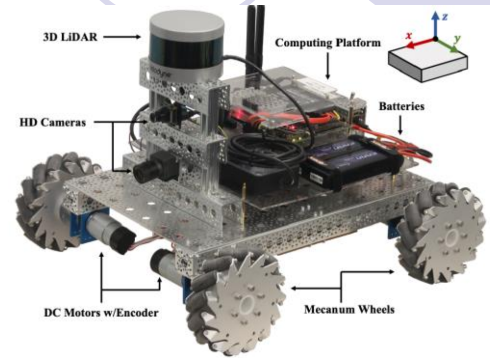
- Combine forward/backward passes (insert 1' after 1 to fill up space)



Duan and Wu, Optimizing Resource Allocation in Pipeline Parallelism for Distributed DNN Training, *Proc. of the IEEE ICPADS*, 2022

An On-going Project

- Extension to DNN training
  - Data compression
- Testbed implementation
  - Visual detection & tracking
- Field test
  - KUSARA at Kettering University



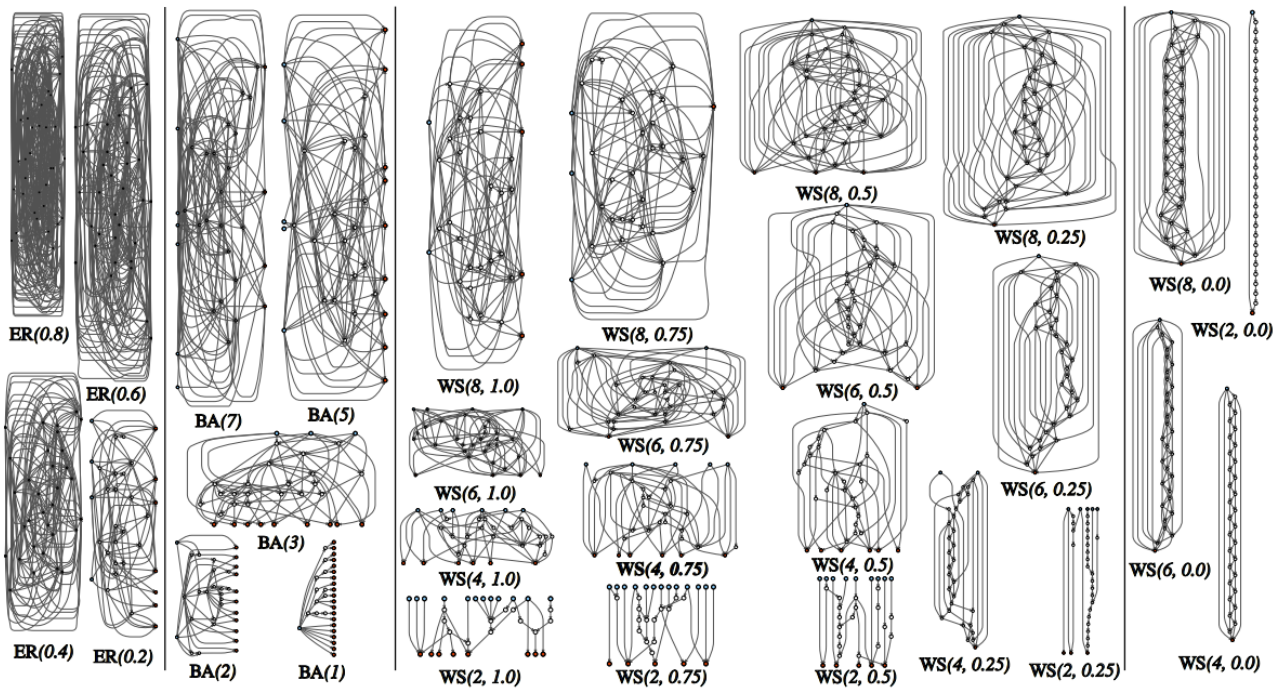
NSF CNS Medium: Cooperative AI Inference in Vehicular Edge Networks for Advanced Driver-Assistance Systems (PI, 2021-2024)

(with Stony Brook, Rowan, and Kettering)

Some Reflections

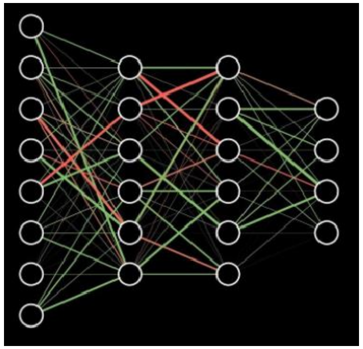
Back to the past: interconnection networks

- Randomly wired NNs (random graphs): neuroscience
- Erdos-Renyi (ER): random, Barabasi-Albert (BA): preferential
- Watts-Strogatz (WS): small-world



Xie et al, Exploring Randomly Wired Neural Networks for Image Recognition, *Proc. of ICCV, 2019.*

5. Conclusions and Future Work



- Offloading as a service
  - Mobile Cloud Computing (MCC)
  - DNN: Single-path, multi-path, and DAG
- Joint partition and scheduling
  - Johnson's rule and its extensions on pipelines
  - Unique properties of comp. and comm. of DNNs
- Future work
  - Optimal partition and scheduling of DAG
  - Pipeline of transfer learning with freeze stage
  - Dynamic nature of offloading speed

Questions

Collaborators: Yubin Duan (Facebook)
Ning Wang (Rowan U.)