An Online Approach for DNN Model Caching and Processor Allocation in Edge Computing

Zhiqi Chen, Sheng Zhang, Zhi Ma, Shuai Zhang, Zhuzhong Qian, Mingjun Xiao, Jie Wu, Sanglu Lu

State Key Lab. for Novel Software Technology

Nanjing University, China

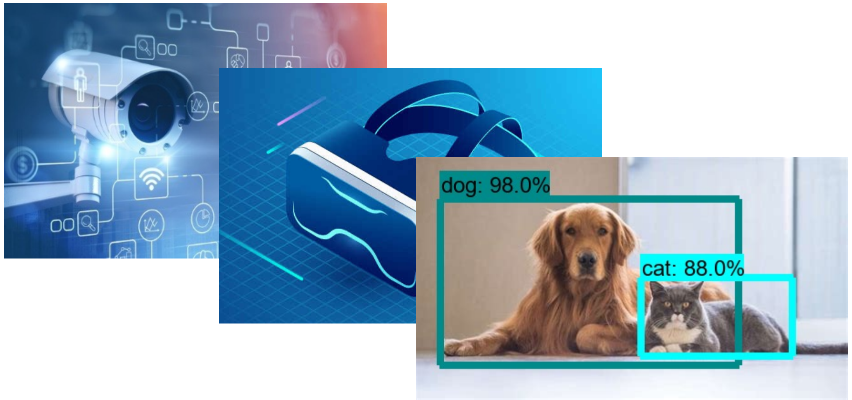
2022/5/17

Outlines

1. Introduction
2. Contributions
3. Problem Formulation
4. Algorithm Design
5. Evaluation
6. Conclusion

1. Introduction

- Edge Computing
  - object detection
  - virtual reality
  - intelligent cameras



- Deep Neural Networks (DNN) inference
  - VGG, ResNet
- Cache DNN models on the edge brings benefits
  - efficiency
  - privacy
  - security

1. Introduction

- Motivation
  - The difference between cloud and edge
    - Cloud: high computing capacity, long transmission delay
    - Edge: short transmission delay, more caching DNN cost
  - Computing is fine-grained on the edge
    - The number of processors assigned to each service will affect the service delay
  - The user request distribution for mobility
    - The user connection information that counts the history on each server can be obtained

2. Contributions

- Achieve the trade-off between user perception delay and energy consumption cost
  - consider DNN model caching and processor allocation in EC
  - NP-Complete
- Propose a novel online algorithm called DMCPA-GS-Online
  - leverages the Lyapunov framework
  - expanded edition of Gibbs Sampling
  - near-optimal
- Evaluate the performance of DMCPA-GS-Online
  - a trace dataset from the real world

3. Problem Formulation

- A user request will encounter two caching hit conditions:
  - Edge-Hit： directly processed by the edge
  - Cloud-Hit： forwarded to the cloud server

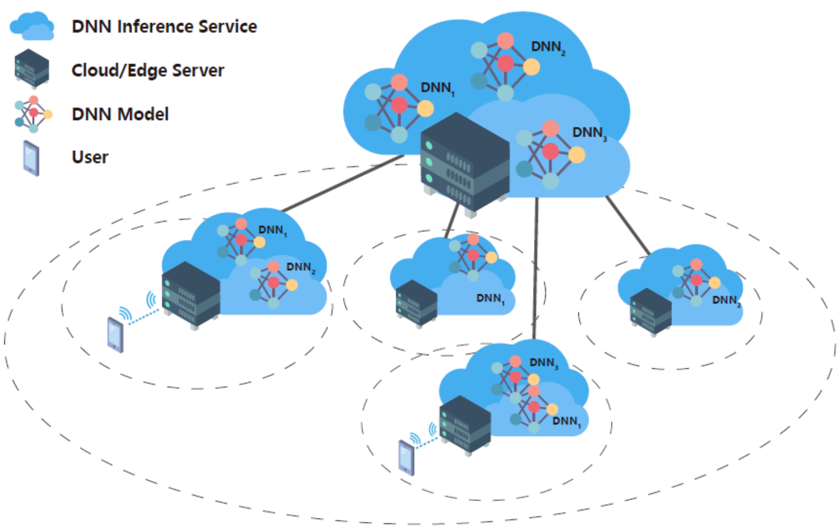


Fig. 1. An overview of our problem.

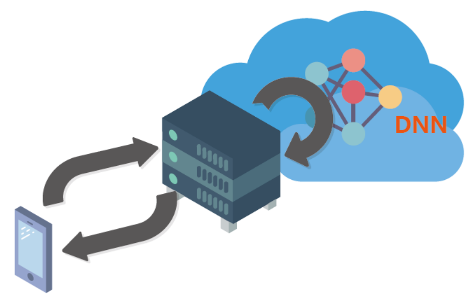


Fig. 2. Edge-Hit

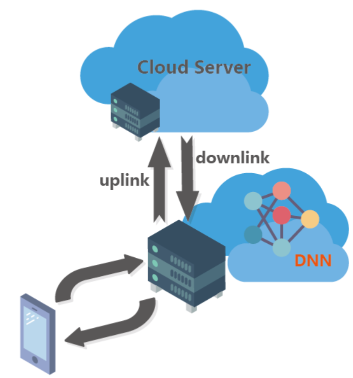


Fig. 3. Cloud-Hit

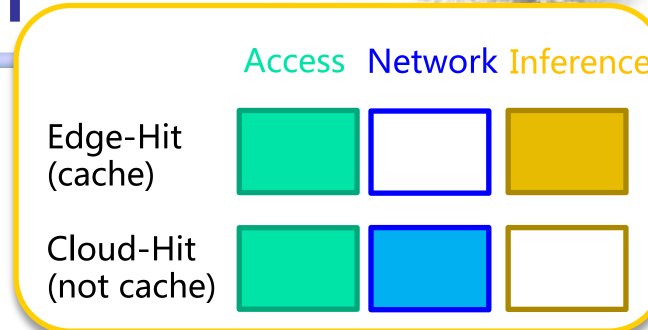
$$x_{i,j}(t) \in \{0,1\}$$

cache decision vector

$$y_{i,j}(t) \in \mathbb{Z}$$

processor allocation vector

3. Problem Formulation



- User Perception Delay
  - Server Access Delay $L^{acc}_{k,i}$
  - Network Transmission Delay
    - cache: 0
    - not cache: $L^{cloud}_{k,i} = \frac{d^{input}_{k}}{B^{cloud}_{i}} + L^{cloud\downarrow}_{k,i}$
  - Model Inference Delay
    - cache: $L^{edge}_{k,i,j} = \frac{\theta d^{input}_{k}(t)}{y_{i,j}(t) c_{i,j}}$
    - not cache: 0 or a constant
- Overall:

$$L_{k,i,j}(t) = L^{acc}_{k,i} + (1 - x_{i,j}(t)) L^{cloud}_{k,i} + x_{i,j}(t) + L^{edge}_{k,i,j}$$

$$L^{P}_{j}(t) = \frac{1}{N_u} \sum_{U_k \in \mathrm{U}} \sum_{S_i \in \mathrm{S}} P_{k,i} L_{k,i,j}(t)$$

← introduce the user request distribution $P_{k,i}$

$$\mathrm{T}^{d}(t) = \sum_{M_j \in \mathrm{M}} \max\{L^{P}_{j}(t) - D_j, 0\}$$

← violation punishment of QoS

3. Problem Formulation

- Energy Consumption Cost
  - The consumption cost to maintain processors: $e_{i,j}$
  - The initialization cost for DNN model: a_j
- Overall

$$\mathrm{T}^{e}(t)=\sum_{S_i\in \mathrm{S}}\sum_{M_j\in \mathrm{M}} e_{i,j}y_{i,j}+a_j \max\{x_{i,j}(t)-x_{i,j}(t-1),0\}$$

- The Total Cost

$$\mathrm{T}(t)=\alpha \mathrm{T}^{d}(t)+\beta \mathrm{T}^{e}(t)$$

3. Problem Formulation

- The Formal Definition of DMCPA

$$\mathcal{P}^0: \min_{\forall t, x(t), y(t)} \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\mathcal{T}(t)]$$

$$
\begin{aligned}
s.t. \quad & C_1: \quad \mathcal{T}(t) = \alpha \mathcal{T}^d(t) + \beta \mathcal{T}^e(t), \\
& C_2: \quad \sum_{M_j \in \mathbb{M}} x_{i,j}(t) w_j \le W_i, \forall S_i \in \mathbb{S}, \\
& C_3: \quad \sum_{M_j \in \mathbb{M}} y_{i,j}(t) \le \phi_i, \forall S_i \in \mathbb{S}, \\
& C_4: \quad x_{i,j}(t) \in \{0,1\}, \forall S_i \in \mathbb{S}, \forall M_j \in \mathbb{M}, \\
& C_5: \quad y_{i,j}(t) \in \mathbb{Z}, \forall S_i \in \mathbb{S}, \forall M_j \in \mathbb{M}, \\
& C_6: \quad \min\{y_{i,j}(t), 0\} \le x_{i,j}(t) \le y_{i,j}(t), \\
& C_7: \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[L_j^P(t)] \le D_j, \forall M_j \in \mathbb{M}.
\end{aligned}
$$

NP-Complete

This is a long-term average optimization problem

↓

use Lyapunov framework

4. Algorithm Design

- Define the queues in each time slot:

$$Q_j(t+1) = \max\{Q_j(t) + L_j^P(t) - D_j, 0\}$$

- Define the quadratic Lyapunov function:

$$L(\Theta(t)) = \frac{1}{2}\sum_j Q_j(t)^2, \quad \text{let } \Theta(t) = [Q_1(t), \cdots, Q_{N_m}(t)]$$

- Define the Lyapunov drift:

$$\Delta(\Theta(t)) = L(\Theta(t+1)) - L(\Theta(t))$$

Using Lyapunov drift, the drift-plus-penalty algorithm can be used, for the fact:

$$\Delta(\Theta(t)) \le B + \sum_j Q_j(t)(L_j^{P,Q}(t) - D_j)$$

4. Algorithm Design

- The origin problem can be converted into a series of subproblems as:

$$\mathcal{P}^3: \min_{\forall t, x(t), y(t)} \quad \mathbb{E}[B + V \cdot \mathcal{T}(t) + \sum_j Q_j(t)(L_j^P(t) - D_j)|\Theta(t)]$$
$$s.t. \quad (C_1) - (C_6),$$

one time slot problem

Algorithm 1: The DMCPA-GS-Online Algorithm

Input: $Q_j(0) \leftarrow 0, x^{prev}(0) \leftarrow 0$
for $t = 0$ *to* T **do**
  receive $d_k^{input}(t)$ from environment;
  update current distribution P from environment;
  get $x^*(t)$, $y^*(t)$ by solving $\mathcal{P}^3$ using Alg. 2;
  $x^{prev}(t+1) \leftarrow x^*(t)$;
  for $M_j \in \mathbb{M}$ **do**
    $Q_j(t+1) \leftarrow \max\{Q_j(t) + L_j^P(t) - D_j, 0\}$;
  end
end

The algorithm for the subproblem

Based on Gibbs Sampling

4. Algorithm Design

- The expanded edition of Gibbs Sampling Algorithm for the subproblem

1. initialize y randomly at the beginning
2. compute the optimal caching strategy x based on y
3. jump from (x, y) to $(x^{'}, y^{'})$ with the probability:

$$\Pr = \frac{1}{1+e^{(T^{'}-T)/\omega}}$$

The probability of getting the global optimal value close to 1 when w is close to 0

proved

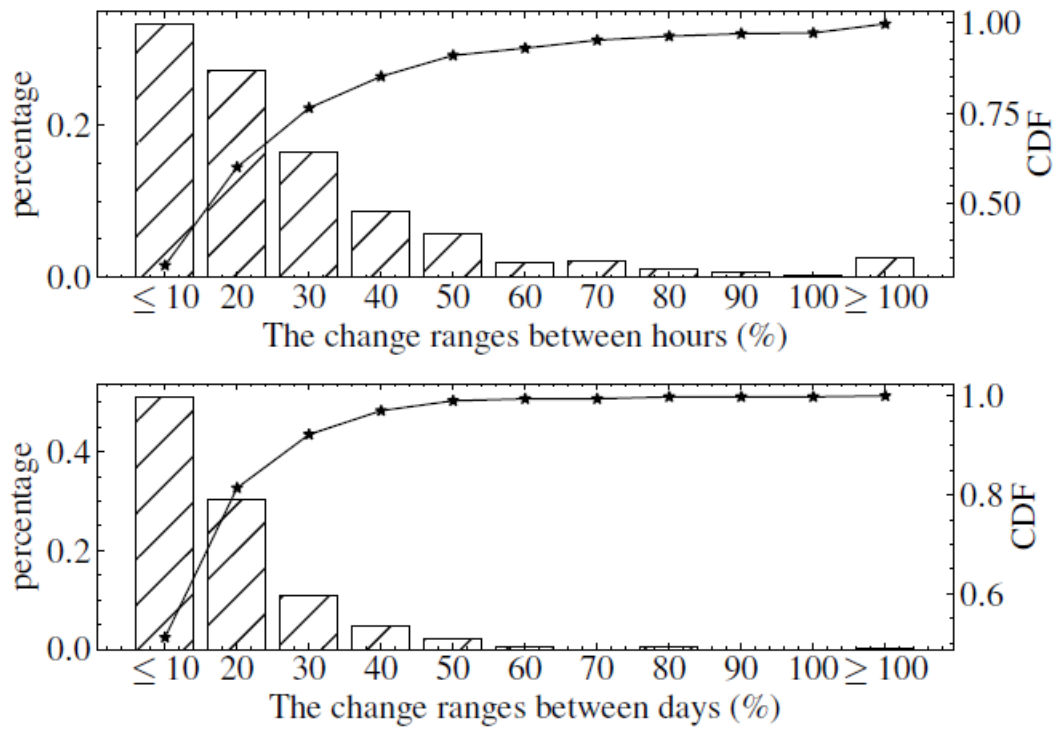
4. repeat jump until converging

5. Evaluation

- Evaluation Setup
  - 5 edge servers
  - at least 200 users
  - P follows the trace of a dataset from the real world
  - d_k^{input} follows Poisson distribution with expectation 100
  - ϕ_i is set to 20
  - w_j follows $N(10,3)$, W_i is 300
  - set $\alpha = \beta = 0.5$

5. Evaluation

- Motivation

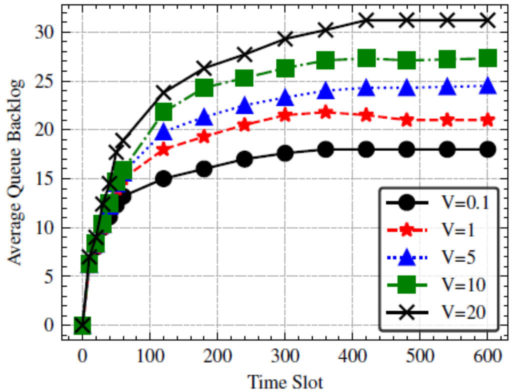


(a) The probability variation.

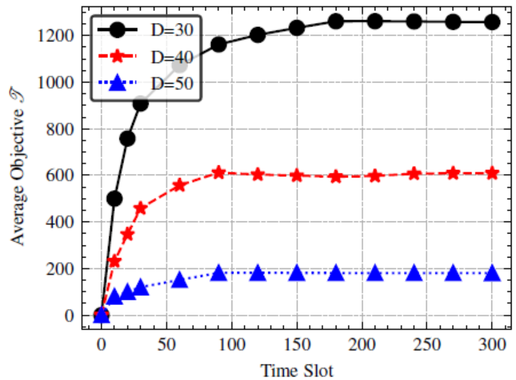
Dataset from the real world

P is stable

5. Evaluation



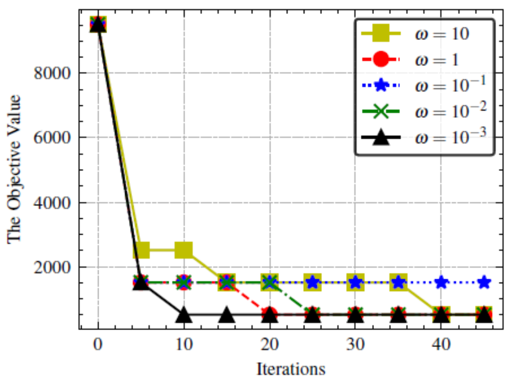
(b) The impact of V.



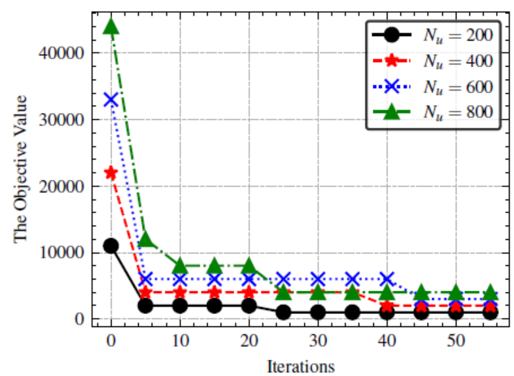
(c) The impact of D_j.

The impact of Lyapunov's parameter

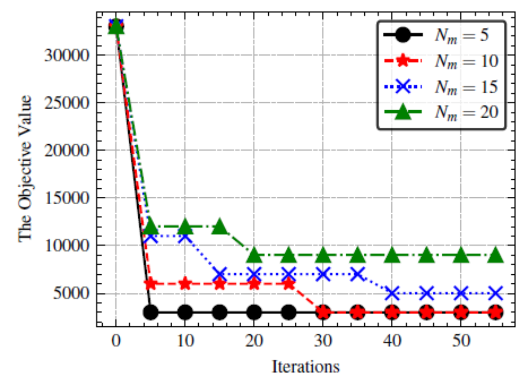
The impact of Gibbs Sampling's parameter



(a) The impact of ω.



(b) The impact of user number.

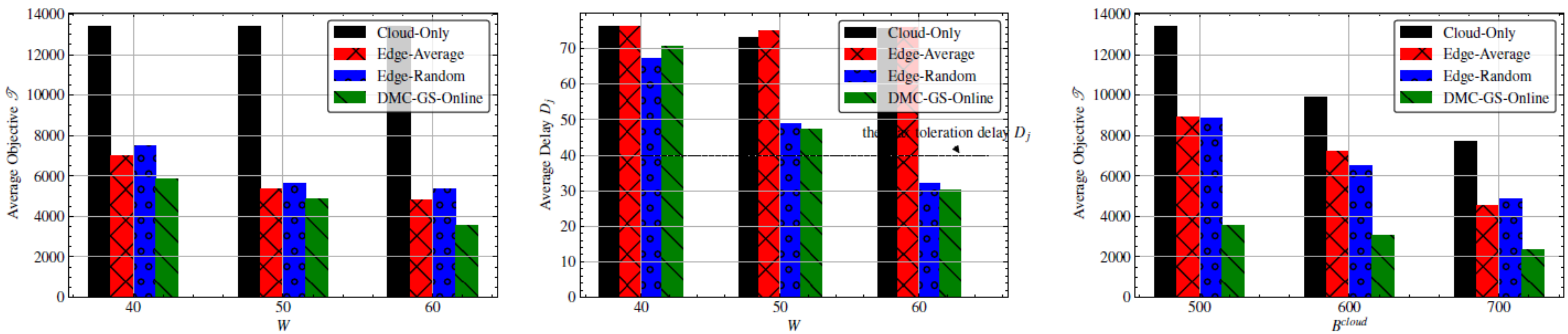


(c) The impact of model number.

Fig. 5. The impact of different parameters for DMCPA-GS.

5. Evaluation

- Cloud-Only
- Edge-Average
- Edge-Random



(a) Average Objective $\mathcal{T}$. (b) Average Delay D_j. (c) Bandwidth B_i^{cloud}.

Fig. 6. The performance of different algorithms.

6. Conclusion

- We consider the DNN Model Caching and Processor Allocation problem.
- Our goal is to minimize user perception delay and energy consumption with careful model caching and processor allocation strategy.
- We formulate it as an Integer Nonlinear Program and the novel online algorithm DMCPA-GS-Online is proposed without future information.
- Experiments based on trace dataset from the real world demonstrate our algorithm outperforms other baselines.

Thanks!

Q&A