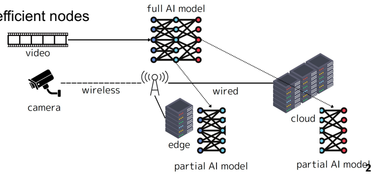
POWER EFFICIENT EDGE-CLOUD COOPERATION BY VALUE-SENSITIVE BAYESIAN ATTRACTOR MODEL

Tatsuya Otoshi, Hideyuki Shimonishi,
Tetsuya Shimonkawa, Masayuki Murata

Osaka University, Japan

Edge-Cloud Cooperated Video Recognition

- By deploying AI at both the edge and in the cloud, and appropriately allocating processing, we can optimize latency, accuracy, and power consumption
- If a certain level of accuracy is acceptable:
  - Process rapidly using lightweight edge computing
  - Reduce power consumption by simplifying processing
- If a certain level of latency is tolerable:
  - Allocate processing to distant but power-efficient nodes



Dividing Workload

- Data Division
  - For models that take images as input, such as Yolo, it is possible to distribute processing on a per-frame basis between edge and cloud.
  - For models like ViViT that take the temporal axis into account, higher accuracy is expected, but the sequence of frames is crucial, making it difficult to divide on a per-frame basis.
- Model Division
  - Divide the model into initial and subsequent processing stages.
  - Even if the input is video, the division does not affect accuracy.

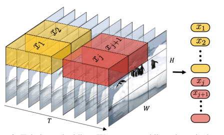


Figure 3: Tubelet embedding. We extract and linearly embed non-overlapping tubelets that span the spatio-temporal input volume.

Arnab, Anurag, et al. "Vivit: A video vision transformer." *Proceedings of the IEEE/CVF international conference on computer vision.* 2021.

Two-step Processing in Edge-Cloud with Divided Model

- System
  - Deploy the initial processing model at the edge, and the subsequent processing model in the cloud.
  - Both edge and cloud connect their outputs to classifiers to perform classification.
  - If the classification at the edge does not yield certainty, the subsequent processing is carried out in the cloud for classification.
- Reduction in Power Consumption
  - If the judgment is completed at the edge alone, cloud processing can be skipped, reducing the associated power consumption.
  - The lighter the model placed at the edge, the greater the benefit when decisions are made at the edge only.
  - However, lighter models tend to be less accurate, which means there's a higher likelihood of needing to proceed to cloud judgment.
  - → Adjust the model at the edge to minimize power consumption.

Formulation of Power Consumption Minimization Problem

$minimize: P = \sum_{x,s}\{P_e(x_s, m_s) + c(x_s, m_s)(P_c(x_s, m'_s) + P_{ec}(x_s, m_s))\} + P_{const}$

$s.t.:$

$D(x_s, m_s) = D_e(x_s, m_s) + c(x_s, m_s)\big(D_c(x_s, m'_s) + D_{ec}(x_s, m_s)\big) < D$

$c(x_s, m_s) = 1_{\{a: a<c\}}\big(A_e(x_s, m_s)\big)$

$A(x_s, m_s) = \big(1 - c(x_s, m_s)\big)A_e(x_s, m_s) + c(x_s, m_s)A_e(x_s, m_s) > A$

s: session
x: video input
m, m': divided models
P_e, P_c, P_{ec}: power consumptions
D_e, D_c, D_{ec}: delays
D: upper bound of delay
A: lower bound of accuracy
$c(x, m)$: using cloud or not
$1_X(x)$: indicator function

- Probabilistic Optimization
  - The input x to the model is a random variable, and power consumption, latency, and accuracy are all random variables.
  - Minimize the expected value of power consumption.
  - For latency, replace the upper limit for the expected value + nσ with D.
  - Optimize with respect to the distribution of input x.
  - In situations where classification is easy, place a small model at the edge to reduce processing at the edge.
  - In difficult classification situations, such as in a crowded space, place a larger model at the edge to reduce data transfer to the cloud.

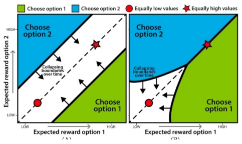
Approach

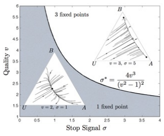
- Adaptation to environmental variations caused by the input video data is necessary.
  - In cases where classification is easy, low power consumption is achieved by classifying with only the lightweight model placed at the edge.
  - When classification is difficult, placing a moderately large model at the edge improves the edge's classification performance.
- To follow fluctuations, it is necessary to transition to the appropriate settings in a short time.
  - There is a possibility that circumstances may change during optimization calculations.
  - It is necessary to make appropriate choices with limited samples for stochastic observations.
- Apply the solution to the Speed-Accuracy Tradeoff in decision making.
  - Value sensitivity: Adjust the speed of choice according to the overall value of the options.
  - Transition settings quickly when the gain in low power consumption is significant.
  - Search for the optimal setting over time when the gain in low power consumption is small.

Value Sensitivity

[1] Pirrone, Angelo, et al. "Magnitude-sensitivity: rethinking decision-making." *Trends in cognitive sciences* 26.1 (2022): 66-80.
[2] Pais, Darren, et al. "A mechanism for value-sensitive decision-making." *PloS one* 8.9 (2013): e73216.

- Overview
  - Value sensitivity refers to the tendency to make choices based on the sum of the values of alternatives (magnitude).
  - When alternatives have high magnitude, accuracy is sacrificed to speed up the decision-making process.
- Advantages
  - It allows waiting for a better choice in the future when alternatives have low magnitude.
  - Value sensitivity plays an important role in consensus building in group decision-making.

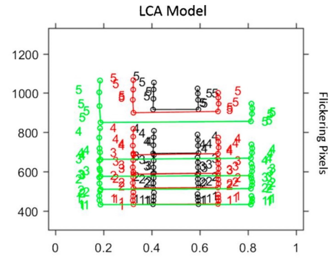




Value Sensitive Models

- Leaky Competing Accumulator (LCA)
  - Drift term depends on the value
  - The value accelerate the state change
  - $z_{t+1}[i] = (1-\gamma)z_t[i] \underbrace{- \beta \sum_{j\neq i} z_t[j]}_{\text{inhibition}} + \underbrace{u_t[i]}_{\text{drift}}$

z: state
u: value
ϵ: noise



[3] Teodorescu, Andrei R., Rani Moran, and Marius Usher. "Absolutely relative or relatively absolute: violations of value invariance in human decision making." *Psychonomic bulletin & review* 23.1 (2016): 22-38.
[4] Ratcliff, Roger, Chelsea Voskuilen, and Andrei Teodorescu. "Modeling 2-alternative forced-choice tasks: Accounting for both magnitude and difference effects." Cognitive psychology 103 (2018): 1-22.

Bayesian Attractor Model (BAM)

- Original BAM
  - Involves Bayesian updating based on observed values
  - Uses attractors and representative values
  - Generative model equations:
    - $z_t = f(z_{t-1}) + qw_t$
    - $x_t = (\mu_1, \cdots, \mu_K)\sigma(z_t) + sv_t$
- Value-Based BAM (VSBAM)
  - Observed and representative values changed to values of alternatives
  - Value estimation obtained through reward feedback information
  - Finds highest-value alternative using recognition scheme
  - Representative values:
    - $\mu_i = (0, \cdots, u_{max}, \cdots, 0)$

z: state
x: observation
μ_i: representative value
v, w: noise
q, s: dynamic-, sensory uncertainty
u_{max}: maximum value

VSBAM-LCA

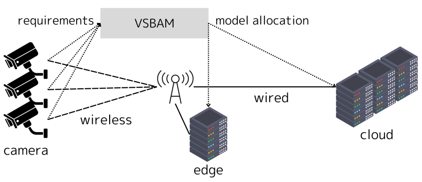
- Update Rule of the Original BAM
  - Drift Term: Posterior distribution update based on likelihood (closeness between representative value and observed value)
  - Noise Term: Normal noise
  - Inhibition Term: Hopfield dynamics

LCA

- Reflecting Value in the Drift Term
  - Update likelihood x value as new likelihood (manipulate the Kalman gain with value)
    - $z_{t+1} = z_t + \frac{u}{\bar{u}} K(x_{t+1} - \hat{x}_{t+1})$
  - Since Kalman gain is almost inversely proportional to sensory uncertainty, it can be considered as manipulating sensory uncertainty according to value
    - $s \to \frac{s}{u_t}$

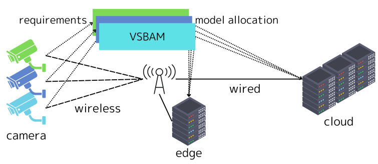
Centralized Model Allocation with VSBAM-LCA

- VSBAM decides on the allocation for all sessions.
  - Combinations of allocations become the options.
  - As the number of sessions increases, the combinations become vast and do not scale.
- Observations:
  - Value of the objective functior
- Options:
  - Allocation for all sessions.



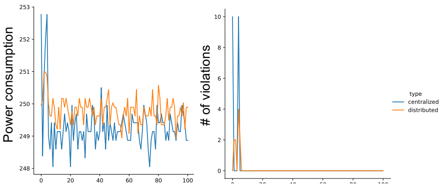
Distributed Model Allocation with VSBAM-LCA

- Deploy individual VSBAM for each session to decide its allocation.
  - Even if the number of sessions increases, the number of options within each VSBAM remains constant.
  - Address inter-session arbitration in a value-sensitive manner.
- Observations:
  - Value of the objective function (global).
- Options:
  - Allocation for each session.



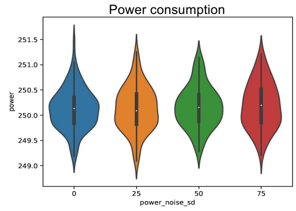
Comparison among Centralized and Distributed VSBAMs

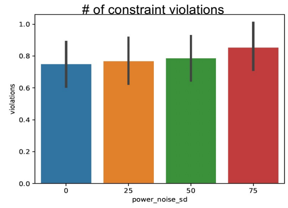
- Settings
  - Number of sessions: 10
  - Latency constraint: 1/15 seconds
  - Centralized: Controlled by a global BAM with combinations of options for each session as attractors
  - Distributed: Controlled by an individual BAM for each session with options for each session as attractors
- Results
  - The centralized approach makes slightly lower power-consuming choices
  - Both distributed and centralized immediately meet the constraints



Robustness to Model Error in Power Consumption

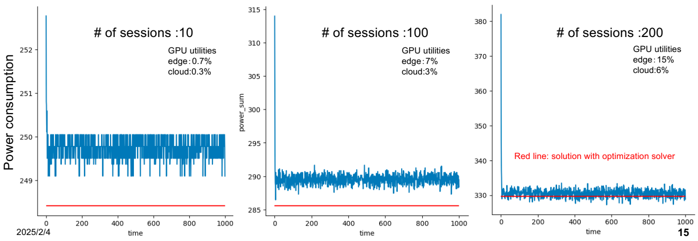
- Evaluate the robustness against errors in the model that predicts power consumption.
- There is no impact on the power consumption itself, with a slight increase in the number of times constraints are violated (less than once in 100-time steps).





Effect of Number of Sessions

- Increasing the number of sessions results in a higher overall computational and communication load, leading to a greater reduction in power consumption.
- When the number of sessions is increased, it approaches the optimal solution



Summary

- Summary
  - We proposed a method for reducing power consumption in environments where AI models are divided into initial and subsequent stages and deployed at the edge and cloud.
  - The proposed method achieves responsiveness to fluctuations and convergence in a distributed environment by applying value sensitivity.
  - It also demonstrated noise tolerance.
- Future Challenges
  - Evaluation in continuous time with repeated environmental changes.
  - Comparison with other distributed optimization methods.
  - Assessment of the impact of different methods of data division and model division.

Thank you for your attention