SETUP

The setup consists of an Edge device, which is resource constrained device, like an IOT/microcontroller and an Edge Server.

Edge Device

- Resource Constrained
- Limited computational Power
- Limited Storage
- Lower inference accuracy for ML/DL models

Edge Server

- Higher storage capacity
- Higher computational power
- Hosts a model with higher inference accuracy

The Edge device collects data from surroundings. Since the SM is trained with a lower number of parameters, its inference accuracy is lower compared to the LM, which is trained with a higher number of parameters.

The edge device can make communicate with the edge server via a wireless network during which the collect data can be forwarded to the edge erver.

Ways in which inferences can be made

- SM(no offloading)
- LM(full offloading)
- Hierarchical Inference(Some offloading)

The data samples are received at the SM. According to the inferences made by the SM, the algorithm decides to accept the inference or offload the data to the LM. The way this works out is the idea of this project.

Using Hierarchical Inference one can achieve the accuracy of LM on a small device

For this purpose, let cost of offloading be some 0<C<I, where I is the cost of incorrect inference at the SM. Let the cost for correct inference of the SM be S.

A simple approach is to offload if the maximum probability among all the classes is less than a threshold, then simply offload the corresponding data samples otherwise not.

So if the accuracy of the SM inference is A then roughly the cost is A*S + I*(1-A) per 100 inferences.

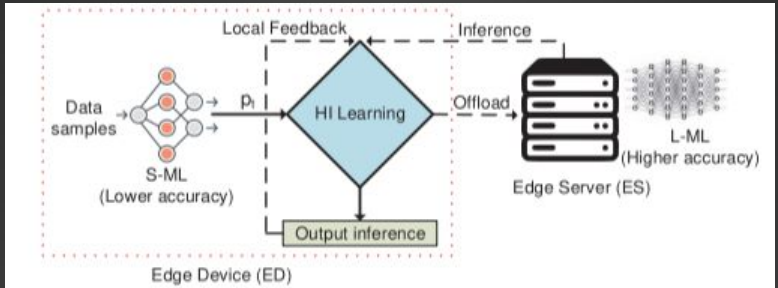


FIGURE 1. Schematic of the HI meta-learning framework.

The threshold can be conveniently chosen if the specific costs are known. For a histogram representing the number of images and the corresponding output probabilities. The probability for which the number of incorrectly and correctly classified images are same if chosen as the threshold probability.

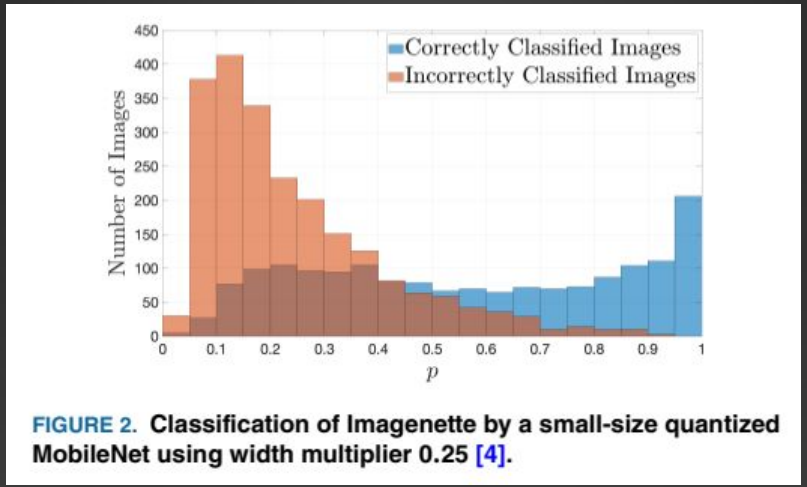


FIGURE 2. Classification of Imagenette by a small-size quantized MobileNet using width multiplier 0.25 [4].

- This problem falls in the domain of Prediction with Experts. However the action space in this case is a continuous so standard cannot be used directly.
- When a data is correctly inferred at the SM then the ground truth is not known at the edge server as the data was never transmitted. This is the case of No-local Feedback HI>
- Another case where the ground truth is always available irrespective of offloading or not is called the Full feedback HI.
- Even in HI, the probability of the maximum probability class is an important metric. Some other parameters like the sub-optimality gap can be integrated for more sophisticated decision making.

The Decision to offload or not is based on the above rule.

A binary random variable Y_t denoting cost of inference, which is 0 for correct inference, 1 otherwise.

$$\mathfrak{D}_t = \begin{cases} \text{Do not offload} & \text{if } p_t \geq \theta_t, \\ \text{Offload} & \text{if } p_t < \theta_t. \end{cases}$$

At any time instance t, the threshold θ_t and probability of maximum probability class p_t, the loss incurred is denoted $l(p_t, \theta_t)$.

Δ_{min} is the minimum suboptimality gap between 2 probability values.

$L(\Theta, Y)$ is the cumulative cost over time n.

Online Learning for Hierarchical Inference

Given a sequence $Y = \{Y_1, Y_2, \ldots, Y_n\}$, the regret is given by

$R_n = E_\pi[L(\Theta, Y)] - L(\Theta^*, Y)$

where $\Theta^* = \text{argmin} \sum_{t=1} l(\Theta_t, Y_t)$ is the policy threshold that minimizes the cumulative cost over time n.

- The goal is to obtain a sublinear upper bound on the above defined regret for a policy π.

This is similar to the Prediction with experts case where the expert advices now lie in the continuous space [0, 1].

- $\Theta_t \equiv$ experts
- Losses of expert ≡ Losses incurred when threshold Θ_k is chosen

Approach using Online Learning

This is equivalent to the prediction with expert advice case where the losses for each expert(loss for each threshold) after revealed after the end of the time index.

$$
\begin{aligned}
w_{t+1}(\theta) &= e^{-\eta \sum_{\tau=1}^{t} l(\theta, Y_\tau)} \\
&= e^{-\eta \sum_{\tau=1}^{t-1} l(\theta, Y_\tau)} e^{-\eta l(\boldsymbol{\theta}, Y_t)} \\
&= w_t(\theta) e^{-\eta l(\theta, Y_t)}. \\
W_{t+1} &= \int_0^1 w_{t+1}(\theta)\, \mathrm{d}\theta.
\end{aligned}
$$

There are 2 challenges while implementing these steps

1. Calculating the decision thresholds
2. Calculating the integral/summation for cumulative loss

These assigned weights for each threshold give the probability distribution to choose the next threshold.

The decision to offload or not depends on the distance between p_t and Θ_t, so the probability to not offload the data sample can be written as

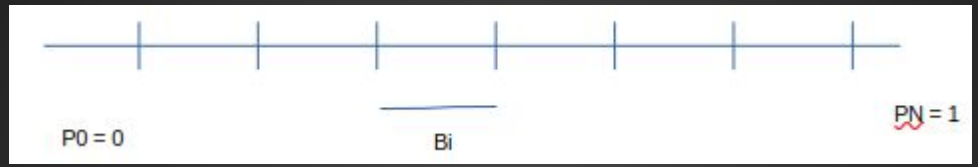
$$q_t = \frac{\int_0^{p_t} w_t(x)\,dx}{W_t}.$$

$w_t(p)/W_t$ is the probability function, and if this lies below p_t, the inferred maximum probability then the data samples are not offloading.

Now the decision module offloads with probability $(1-q_t)$ and does not offload with probability q_t.

This is the first challenge is addressed

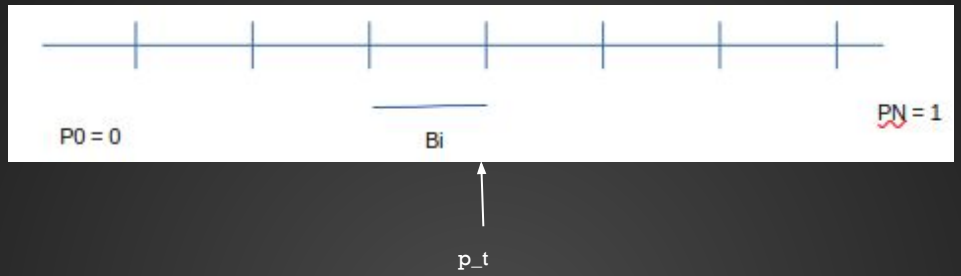
The cumulative loss function $L(\boldsymbol{\theta}_t, \boldsymbol{Y}_t) = \sum_{\tau=1}^{t} l(\theta_\tau, Y_\tau)$, can take 3^t values for the defined cost values {0, C, 1}, without any pattern. So using direct numerical methods for calculating the loss is not possible.



- The line between [0, 1] is divided into N values according to the p_n values.
- The intervals are denoted by $B_i = (p_{[i-1]}, p_{[i]}]$.
- M_i be the number of times $p_{[i]}$ is repeated in p_n.
- $Y_{[i]}$ be the ith local inference cost arranged according to the increasing values of corresponding probability p_i.

  Even if 2 probabilities p_j and p_k are same the inference costs can be different.

For some i, p_t falls on the boundary of some B_i,



$$\Rightarrow l(\theta_t, Y_t) = \begin{cases} Y_t, & \forall\theta : \theta_t \in B_i \subset (0, p_t], \\ \beta, & \forall\theta : \theta_t \in B_i \subset (p_t, 1]. \end{cases}$$

These B_i are chosen such that no two p_t fall in one interval B_i. So for any threshold θ that lies inside interval B_i, the loss is irrespective of the location of θ.

So it can be concluded that the loss function is piecewise constant in each interval B_i.

$$\Rightarrow L(\boldsymbol{\theta}^*, \boldsymbol{Y}) = \min_{\theta \in [0,1]} L(\theta, \boldsymbol{Y}) = \min_{1 \le i \le N} \sum_{t=1}^{n} l(B_i, Y_t).$$

$$\sum_{t=1}^{n} l(B_i, Y_t) = \sum_{t=1}^{n} [\beta \, \mathbb{1}(p_t < p_{[i]}) + Y_t \, \mathbb{1}(p_t \ge p_{[i]})]$$

$$= \beta \sum_{j=1}^{i-1} m_j + \sum_{k=1+\sum_{j=1}^{i-1} m_j}^{n} Y_{[k]}$$

$$\Rightarrow L(\boldsymbol{\theta}^*, \boldsymbol{Y}) = \min_{1 \le i \le N} \left\{ \beta \sum_{j=1}^{i-1} m_j + \sum_{k=1+\sum_{j=1}^{i-1} m_j}^{n} Y_{[k]} \right\}$$

The decision probability for offloading can be calculated by summing the areas of rectangles in these intervals B_i's.

HI Full Feedback

Irrespective of whether the data samples are offloaded or not, the ground truth(LM inference) is always available at the edge device.

The standard EXP3 algorithm(except for continuous action space) can be used.

- After calculating the decision probability, the expected cost in round t can be written as

$$\bar{l}(Y_t) = \mathbb{E}_{Q_t}[l(\theta_t, Y_t)] = Y_t q_t + \beta(1 - q_t),$$

- If p_t is not a repetition then update the intervals by splitting at p_t, then update weights for all intervals.

  The regret for this algorithm, where $L(Y) = \Sigma\, l(Y_t)$

$$R_n = \bar{L}(\boldsymbol{Y}) - L(\boldsymbol{\theta}^*, \boldsymbol{Y}) \leq \frac{1}{\eta} \ln \frac{1}{\lambda_{\min}} + \frac{n\eta}{8}.$$

Algorithm 1 The HIL-F Algorithm for Full Feedback

Initialise: Set $w_1(\theta) = 1, \forall \theta \in [0, 1]$ and $N = 1$.
for every sample in round $t = 1, 2, \ldots$ **do**
    S-ML outputs p_t.
    Compute q_t using (7) and (8), and generate Bernoulli random variable Q_t with $\mathbb{P}(Q_t = 1) = q_t$.
    if $Q_t = 1$ **then**
        Accept the S-ML inference and receive cost Y_t.
    else
        Offload the sample and receive cost β.
    end if
    Find the loss function using (3).
    if p_t is not a repetition **then**
        Update the intervals by splitting the interval containing p_t, at p_t. Increment N by 1.
    end if
    Update the weights for all intervals using (6), based on the interval positions with respect to p_t.
end for

HI No-local Feedback

This is a variant of the Prediction with experts problem, where the costs are not known(if the SM inference is accepted). The ground truth is not available at the edge device.

Exploration is necessary in this case to learn the ground truth. At each time index, the algorithm explores with probability ε irrespective of q_t. Now the cost Y_t can be computed.

Most of the steps in the algorithm are similar except for loss incurred.

Since the loss when not offloaded is not known, a pseudo loss function is used.

$$\tilde{l}(\theta_t, Y_t) = \begin{cases} 0 & p_t \geq \theta_t, Z_t = 0; \ [Do\ Not\ Offload] \\ \dfrac{Y_t}{\epsilon} & p_t \geq \theta_t, Z_t = 1; \ [Offload] \\ \beta & p_t < \theta_t. \qquad [Offload] \end{cases}$$

- The weights update now use the modified pseudo cost function.
- This pseudo cost function is an unbiased estimate of the actual cost function (E[Pseudo cost) = actual cost).
- Regret here can be defined as the difference between the Expected Loss wrt random variables Q and ε and cost incurred for optimal threshold.

$$R_n \leq n\beta\epsilon + \frac{n\eta}{2\epsilon} + \frac{1}{\eta}\ln(1/\lambda_{\min}).$$

Algorithm 2 The HIL-N Algorithm

Initialise: Set $w_1(\theta) = 1, \forall \theta \in [0, 1]$ and $N = 1$.
for $t = 1, 2, \ldots$ **do**
  S-ML outputs p_t.
  Compute q_t using weights from (14) and (15) and generate Bernoulli random variables Q_t and Z_t with $\mathbb{P}(Q_t = 1) = q_t$ and $\mathbb{P}(Z_t = 1) = \epsilon$.
  if $Q_t = 1$ **and** $Z_t = 0$ **then**
    Accept the S-ML inference and receive cost Y_t (unknown).
  else
    Offload the sample and receive cost β.
  end if
  Find the pseudo loss function using (13).
  if p_t is not a repetition **then**
    Update the intervals by splitting the interval containing p_t at p_t. Increment N by 1.
  end if
  Update the weights for all intervals using (14), based on the interval positions with respect to p_t.
end for

$$
\begin{aligned}
&(i)\ j && \leftarrow \max\{i : p_{[i]} < p_t\}.\\
&(ii)\ dup && \leftarrow \textit{FALSE}, \text{ if } p_\tau \neq p_t,\ \forall \tau < t,\ \textit{TRUE} \text{ otherwise.}\\
&(iii)\ q_t && \leftarrow \frac{\sum_{i=1}^{j} w_{i,t}(p_{[i]} - p_{[i-1]}) + w_{j+1,t}(p_t - p_{[i]})}{\sum_{i=1}^{N} w_{i,t}(p_{[i]} - p_{[i-1]})}\\
&(iv)\ N && \leftarrow \begin{cases} N & (dup = \textit{TRUE}),\\ N+1 & (dup = \textit{FALSE}). \end{cases}\\
&(v)\ p_{[i]} && \leftarrow \begin{cases} p_{[i]} & i \le j \text{ or } (dup = \textit{TRUE})\\ p_t & i = j+1 \text{ and } (dup = \textit{FALSE})\\ p_{[i-1]} & j+1 < i \le N \text{ and } (dup = \textit{FALSE}) \end{cases}\\
&(vi)\ w_{i,t} && \leftarrow \begin{cases} w_{i,t-1}e^{-\eta\beta} & p_{[i]} > p_t,\ (dup = \textit{TRUE})\\ w_{i-1,t-1}e^{-\eta\beta} & p_{[i]} > p_t,\ (dup = \textit{FALSE})\\ w_{i,t-1}e^{-\eta Y_t} & p_{[i]} \le p_t, \text{HIL-F}\\ w_{i,t-1}e^{-\eta Y_t/\epsilon} & p_{[i]} \le p_t, Z_t = 1, \text{HIL-N}\\ w_{i,t-1} & p_{[i]} \le p_t, Z_t = 0, \text{HIL-N}. \end{cases}
\end{aligned}
$$