: Prof. Sharayu

Hierarchical Inference

Vinay Sutar, 21d070078

Supervisor

TABLE OF CONTENTS

1. Problem Statement
2. Datasets and Models
3. Initial Approach
4. Algorithm
5. Experimental Results

Problem Statement

- A Edge Device (ED) with a small-size ML model (S-ML) and an Edge Server (ES) with a large-size ML model (L-ML) are used for classification.
- S-ML offers lower accuracy but reduces latency, bandwidth, and energy consumption, while L-ML improves accuracy but increases resource usage.
- Hierarchical Inference (HI) aims to balance the tradeoff by accepting S-ML's inference when correct, offloading incorrect samples to ES.
- Since ED cannot directly know S-ML's correctness, a framework is proposed to predict the correctness based on the output by S-ML and decide whether to offload the data.

The primary task is to implement a decision module for accepting or offloading the SML outputs.

Datasets and Models

Edge Device:

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

Edge Server:

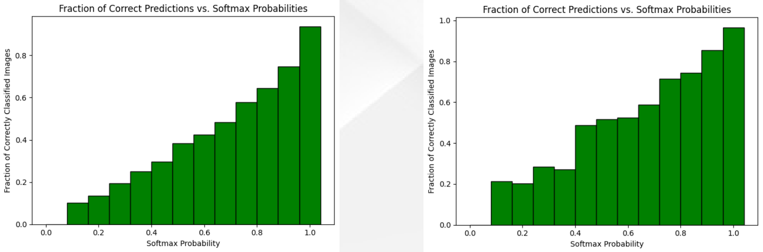
- Higher storage capacity
- Higher computational power
- Hosts a model with higher inference accuracy

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

- **Datasets**: Imagenet-1K and CIFAR100
- **Models**: ResNet50, ResNet18, ShuffleNet-V2, ResNet101, ResNet152

Initial Approach

For the chosen datasets and models above, lets look at the plot of fraction of samples correct for a given output softmax probability.
The output softmax probability refers to the maximum of the softmax values corresponding to the classes.



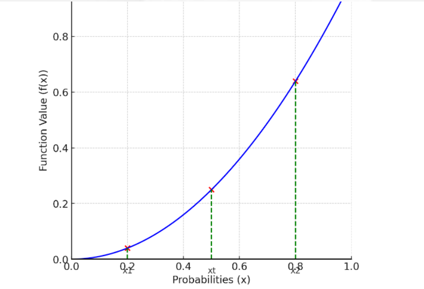
(a) ResNet 101 on CIFAR100

(b) ResNet50 on ImageNet-1K

Figure: Fraction of correctly classified images vs Softmax probability

Approach

With appropriate bin size on the x-axis, it is clear the the fraction of correctly classified images is a non-decreasing function. (Tested over multiple models and datasets).



This can be modeled as: $f(x_t) = \mathbb{P}(\text{inference is correct} \mid \text{confidence} = x_t)$, where the probability is shown on the y axis and f(.) is the non-decreasing function that is used to models this non-decreasing structure.

Approach

Cost structure:

- Offload Cost = τ
- Accept but incorrect inference = 1
- Accept and correct inference = 0

From the definition of f(.), we note the following:

- $f(x_t)$ - Probability of correct inference given confidence is x_t
- $1\text{-}f(x_t)$ - Probability of incorrect inference given confidence is x_t

Thus, the decision rule we adopt is, offload when expected cost of offload is less than expected cost of incorrect offload.

- Expected cost of offload = τ
- Expected cost of incorrect inference = $1.(1 - f(x_t)) + 0.(1 - (1 - f(x_t))) = 1 - f(x_t)$

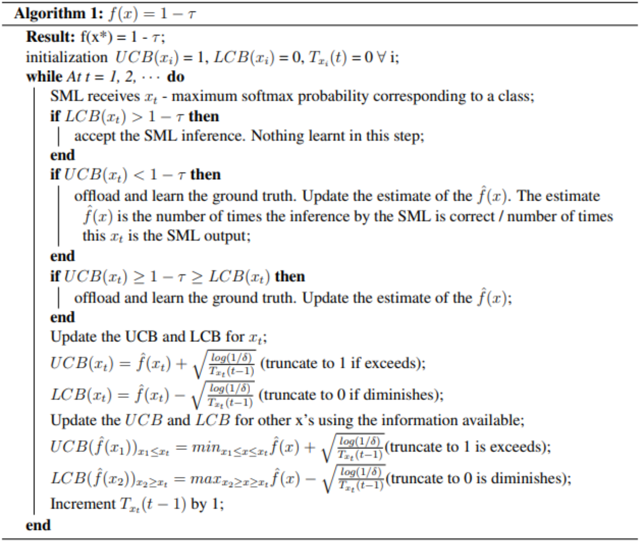
Decision Module

Offload Decision: $\tau < 1 - f(x_t) \implies f(x_t) < 1 - \tau$
The decision, in the end, reduces to finding x where f(.) intersects y = 1 - τ.
To the right of this intersection point (f(x) > 1 - τ) accept the data samples and offload any data samples to the left (f(x) < 1 - τ).

Preliminaries:

- The SML output is converted to a 4 bit value (convenient to keep track).
- The interval [0, 1] is divided into 15 intervals with boundaries generated by these 4 bit values.
- The estimate $\hat{f}(.)$ is maintained for each of these 16 values.

Algorithm



```
Algorithm 1: $f(x) = 1 - \tau$
Result: f(x*) = 1 - $\tau$;
initialization $UCB(x_i) = 1$, $LCB(x_i) = 0$, $T_{x_i}(t) = 0$ ∀ i;
while At t = 1, 2, $\cdots$ do
    SML receives $x_t$ - maximum softmax probability corresponding to a class;
    if $LCB(x_t) > 1 - \tau$ then
        accept the SML inference. Nothing learnt in this step;
    end
    if $UCB(x_t) < 1 - \tau$ then
        offload and learn the ground truth. Update the estimate of the $\hat{f}(x)$. The estimate
          $\hat{f}(x)$ is the number of times the inference by the SML is correct / number of times
          this $x_t$ is the SML output;
    end
    if $UCB(x_t) \geq 1 - \tau \geq LCB(x_t)$ then
        offload and learn the ground truth. Update the estimate of the $\hat{f}(x)$;
    end
    Update the UCB and LCB for $x_t$;
    $UCB(x_t) = \hat{f}(x_t) + \sqrt{\frac{log(1/\delta)}{T_{x_t}(t-1)}}$ (truncate to 1 if exceeds);
    $LCB(x_t) = \hat{f}(x_t) - \sqrt{\frac{log(1/\delta)}{T_{x_t}(t-1)}}$ (truncate to 0 if diminishes);
    Update the $UCB$ and $LCB$ for other x's using the information available;
    $UCB(\hat{f}(x_1))_{x_1 \leq x_t} = min_{x_1 \leq x \leq x_t} \hat{f}(x) + \sqrt{\frac{log(1/\delta)}{T_{x_t}(t-1)}}$(truncate to 1 is exceeds);
    $LCB(\hat{f}(x_2))_{x_2 \geq x_t} = max_{x_2 \geq x \geq x_t} \hat{f}(x) - \sqrt{\frac{log(1/\delta)}{T_{x_t}(t-1)}}$(truncate to 0 is diminishes);
    Increment $T_{x_t}(t-1)$ by 1;
end
```

Figure: Algorithm

Algorithm

For each value/threshold on the x axis, along with $\hat{f}(.)$, a UCB(.) and LCB(.) is maintained as a measure of uncertainty in the estimate of f(.).

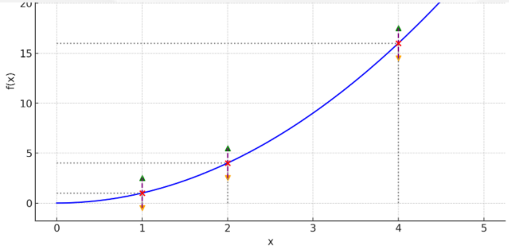


Figure: Structural Property of f(.)

- The data samples whose LCB lies above 1 - τ (to the right) are certainly accepted
- The data samples whose UCB lies above 1 - τ are certainly offloaded
- The data samples for the 1 - τ lie between the LCB and UCB then offload due to ambiguity.

Algorithm

The UCB and LCB for the received probability output is updated as per the standard UCB algorithm (except a modification where the uncertainty is scaled to not exceed 1 initially as the UCB cannot exceed 1).

We can take advantage of the structural property of the function f(.)

- It is evident that UCB of a threshold cannot exceed the UCB of any threshold on its right, so the UCB of some $x_1 < x_t$ at time instant t when the SML output is x_t will be the minimum of the UCB of x, where $x \in [x_1, x_t]$.
- Similarly, LCB of a threshold cannot be lower than the LCB of any threshold on its left, so the LCB of some $x_2 > x_t$ at time instant t when the SML output is x_t will be the maximum of the LCB of x, where $x \in [x_t, x_2]$.

Testing

For testing purposes, the following models and datasets are used:

- ResNet50 with CIFAR100 with an accuracy of 50.14%
- ShuffleNet-V2 with IMageNet-1K with an accuracy of 60.64%

Since LML model is assumed to be always correct, so the ground truth labels are directly used as the LML outputs. However, an offload cost is still considered.

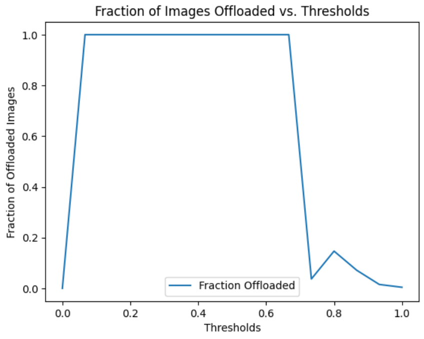


Figure: Fraction of offloaded images vs output probability(ResNet50 with ImageNet-1K_V1)

Testing

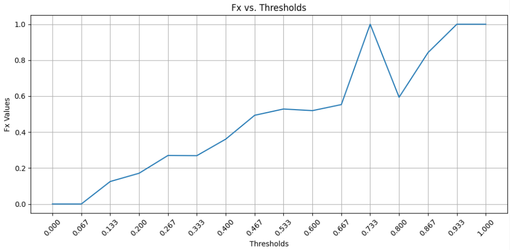


Figure: Plot of estimated $\hat{f}(.)$

The hypothesis of f being a increasing function is validated here.

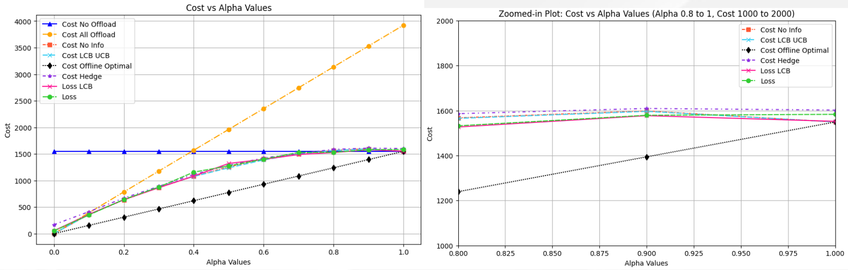
Testing

The performance of these algorithm in terms of cost are compared.

- No Offload - All the inference by the SML are accepted
- All Offload - All the data samples are offloaded irrespective of the SML confidence.
- Offline Optimal - Use the ground truth to offload only the samples for which the inference is incorrect.
- Hedge Algorithm - (Prediction with experts problem) Assign weights to each threshold(expert).
- LCB-UCB algorithm - The algorithm described here is called LCB-UCB algorithm.
- LCB-UCB no info - In this algorithm the 2 steps where the structural information of f(.) is exploited are omitted.
- Min Loss LCB - At any instant t, the threshold with minimum loss until that time is chosen to make decisions.

Comparison (Cost)

The following cost plot vs offload cost τ is obtained.



(a) Cost vs τ

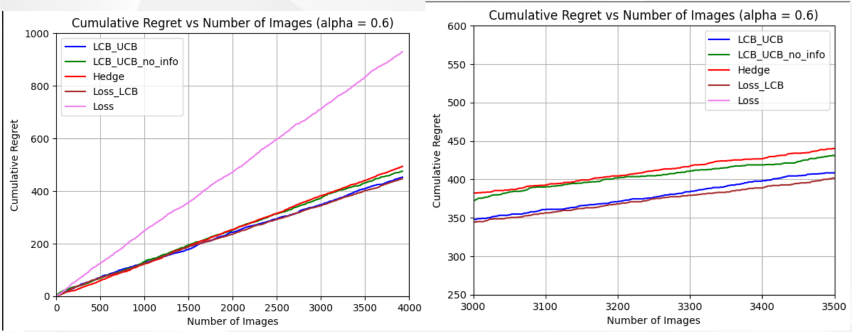
(b) Zoomed in - Cost vs τ

Figure: Cost vs τ for ShuffleNet-V2 with ImageNet-1K

- The no offload algorithm gives a constant cost equal to the (number of incorrect samples)*1
- The all offload algorithm gives a linearly increasing cost with τ
- The offline optimal is the best performance that can be obtained.
- The other algorithm perform approximately similar within a margin of 5%.

Comparison (Regret)

The following regret plot vs offload cost τ is obtained.



(a) Regret vs τ

(b) Zoomed in - Regret vs τ

Figure: Regret vs τ for ShuffleNe-V2 with ImageNet-1K

Comparison (Regret)

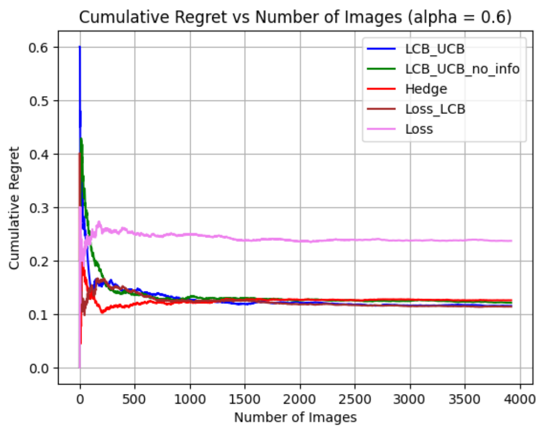
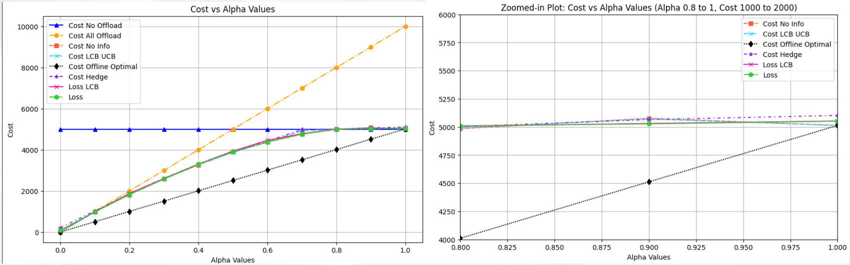


Figure: Cumulative Average regret vs τ for ShuffleNet-V2 with ImageNet-1K

Since the regret varies sublinearly and the number of image increases linearly, the average regret is expected to decrease with time as is validated from the graph above.

Comparison (Cost)

The following cost plot vs offload cost τ is obtained.



(a) Cost vs τ

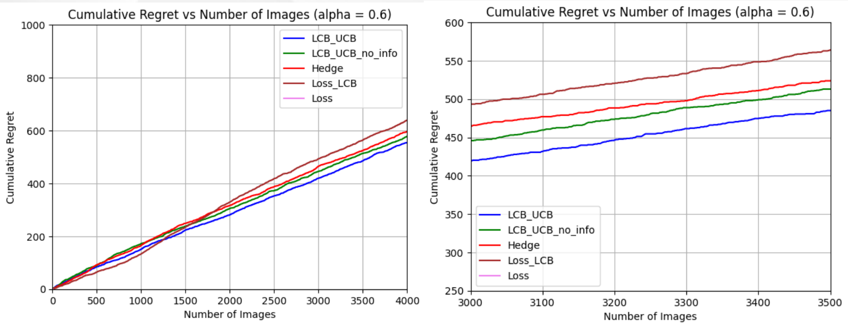
(b) Zoomed in - Cost vs τ

Figure: Cost vs τ for ResNet50 with CIFAR100

- The no offload algorithm gives a constant cost equal to the (number of incorrect samples)*1
- The all offload algorithm gives a linearly increasing cost with τ
- The offline optimal is the best performance that can be obtained.
- The other algorithm perform approximately similar within a margin of 5%.

Comparison (Regret)

The following regret plot vs offload cost τ is obtained.



(a) Regret vs τ

(b) Zoomed in - Regret vs τ

Figure: Regret vs τ for ResNet50 with CIFAR100

Future Work

- Work out the regret analysis bounds for the proposed algorithm.
- Test the consistency of the algorithm over multiple datasets and algorithms
- Dynamically update the hyper parameters while training
- Use multiple layer Hierarchical Inference like device to edge and then edge to cloud
- Instead of offloading the entire data samples, just offload features captured from initial layers of the SML and used these features to capture more intrinsic details.