Early exits and Split computing

MSDNet, SPINN

Simon Lebeaud

GreenAI UPPA

January 24, 2022

Table of Contents

GreenAI U·P·P·A

Table of Contents

MSDNet - Context

- State of the art incentivize resource-hungry models

- Two type of images:
  - "Easy" images, need smaller models for classification
  - "Hard" images, need to go through bigger models

So how do we compromise between those type of image for classification?

GreenAI
U · P · P · A

MSDNet - Problem Setup

Two setting for computational constraints:

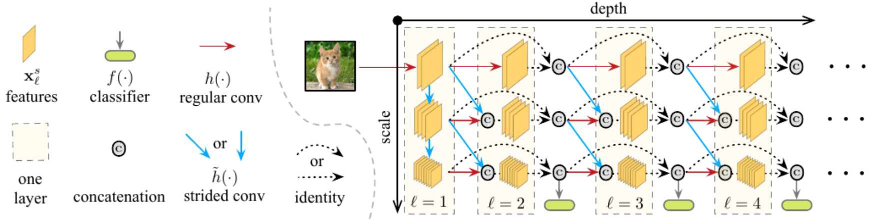
- **Anytime prediction :** finite and nondeterminsitic computational budget $B > 0$ for each images to be classified
- **Budgeted batch classification :** finite computational budget $B > 0$ for a set of D_{test} exemples. Here the model decide how much to spend on each images.

MSDNet - Problem Solving

Two reasons why intermediate ealy exits hurt performance of DNN :

1. **The lack of coarse-level features**
   - Solution : **Multi-scale feature maps**
2. **Early classifiers interfere with later classifiers**
   - Solution : **Dense connectivity**

MSDNet - Architecture



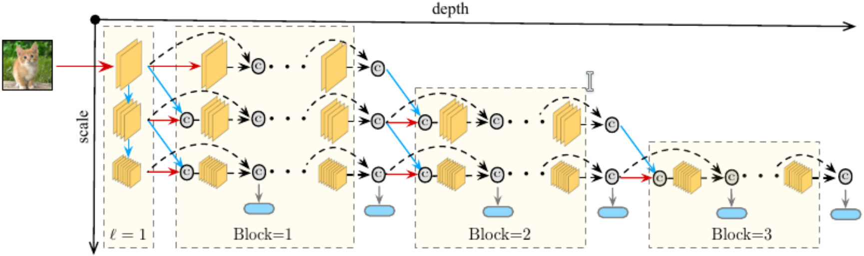
Exit condition :

We have $q_k = z(1-q)^{k-1}q$ with z such that $\sum_k p(q_k) = 1$

We solve for q : $|D_{test}| \sum_k q_k C_k \leq B$

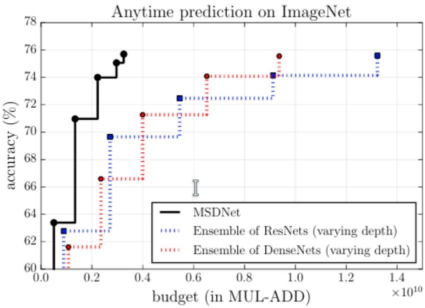
We determine the threshold θ_k on a validation set such that $|D_{test}|q_k$ samples exit at the k_{th} exit.

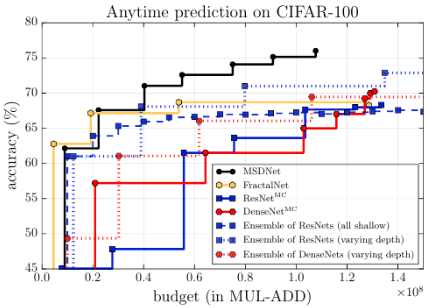
MSDNet - Network reduction



MSDNet - Testing and Results

Anytime prediction:

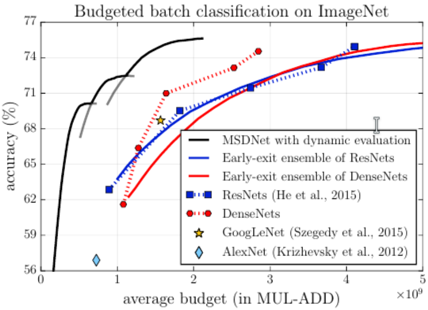




GreenAI
U · P · P · A

MSDNet - Testing and Results

Budgeted batch classification:



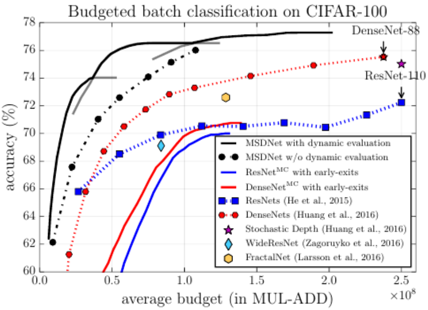
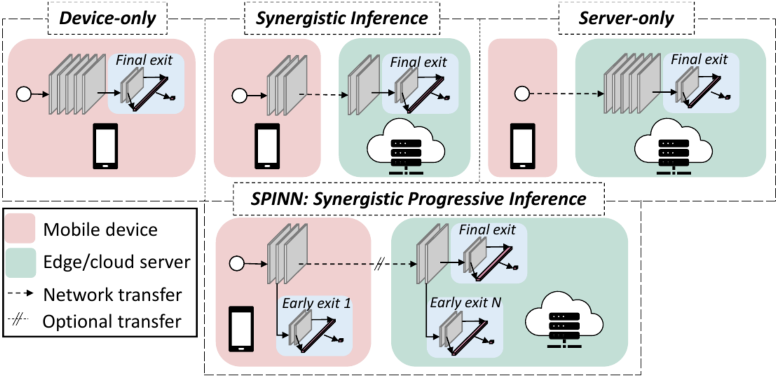
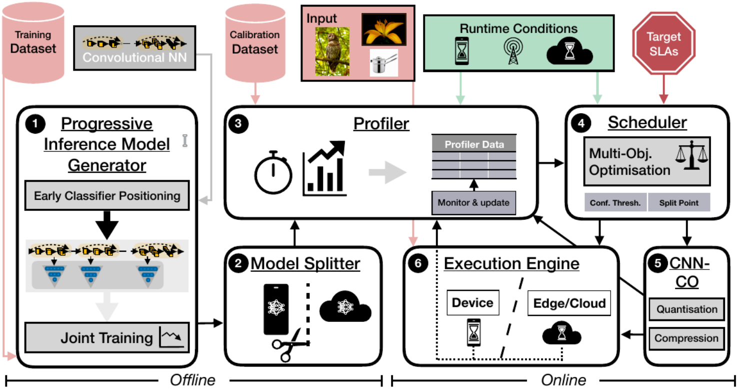


Table of Contents

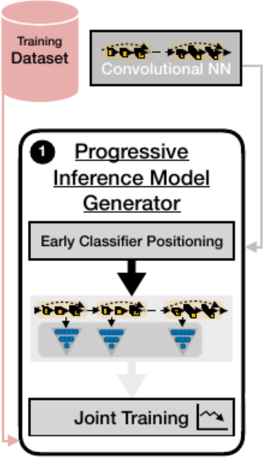
SPINN - Context



SPINN - Architecture



SPINN - Progressive inference model generator



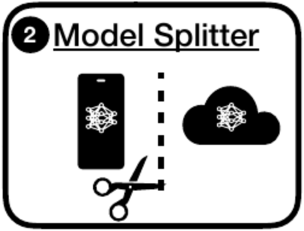
- 6 early exits place equidistantly (15%, 30%,..., 90%)
- training end to end or fine tuning on classifier with frozen backbone if pretrained
- $softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$

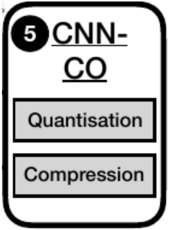
  $arg_i\{\max_i\{softmax_i\} > thr_{conf}\}$

  ${}_{j \in classifier}\{\max_i\{softmax_i^j\}\}$

GreenAI
U · P · P · A

SPINN - Model Spliter and Communication Optimiser



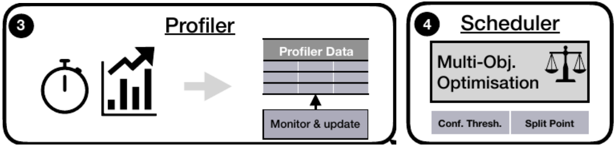


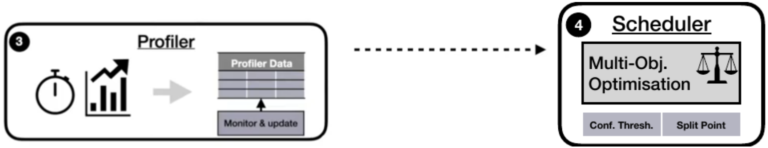
Input: Trained model
Output: Split point candidates

Split at ReLU activation for better packing.

- **Lossy 8 bit Compression**
- **Bit Shuffling**
- **LZ4 Compressions**

SPINN - Profiler Scheduler





Offline component :

- Device-agnostic:
  - Accuracy per exit
  - Size of data to be transmited
- Device-specific:
  - Latency

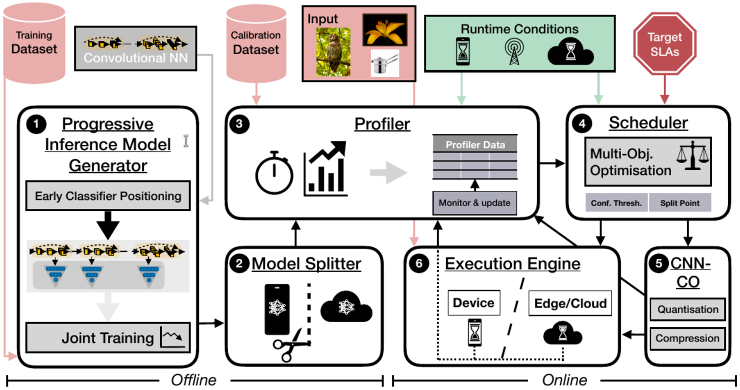
Online component :

- Runtime conditions
- Runtime monitoring

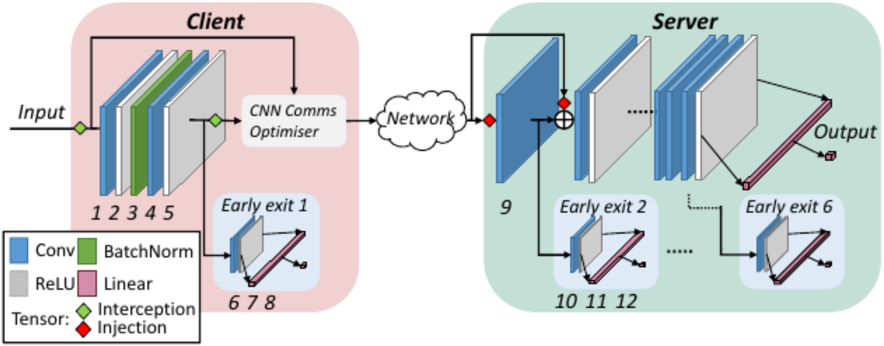
- Removes infeasible points
- Ranks and select best design
- Tunes early exit confidence threshold

GreenAI U · P · P · A

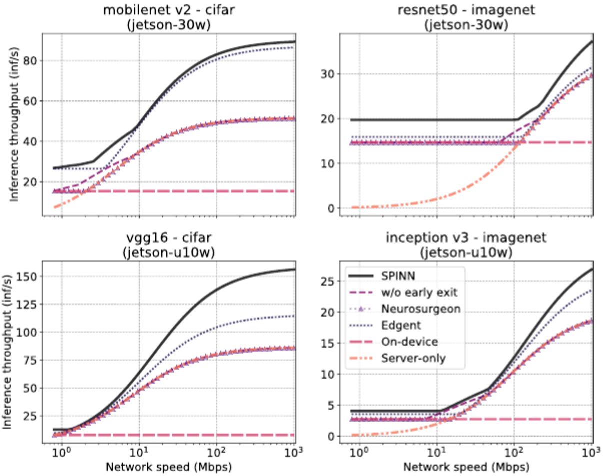
SPINN - Execution



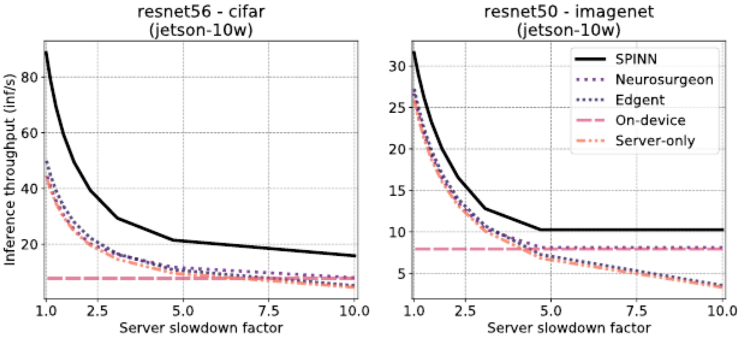
SPINN - Execution



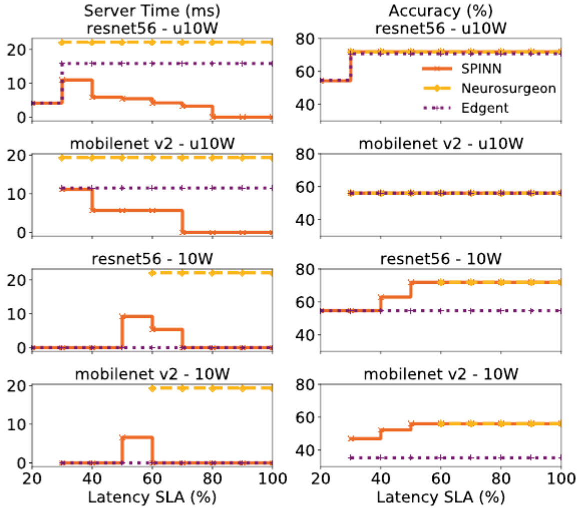
SPINN - Evaluations



SPINN - Evaluations

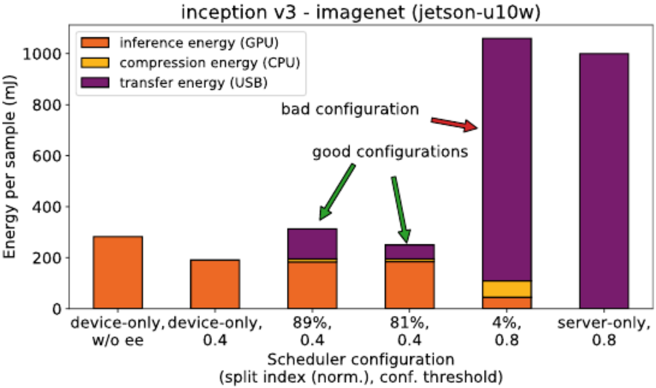


SPINN - Evaluations



GreenAI
U · P · P · A

SPINN - Evaluations



SPINN - Conclusion

SPINN delivers a progressive inference network, that is scalable to **environment conditions** and **app-specific performance goals**:

- **Delivers higher performance** than state of the art,
- Doesn't sacrifices on **accuracy**.