15-884: Machine Learning Systems

TinyML

Instructor: Tianqi Chen

Carnegie Mellon University
School of Computer Science

catalyst

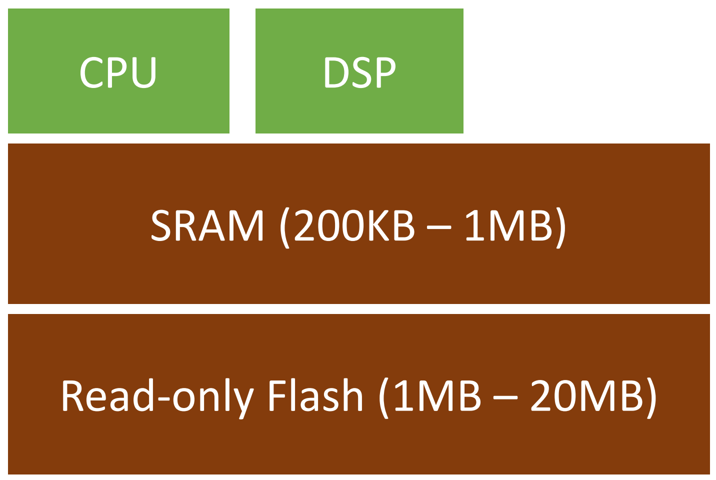
Machine Learning is Getting into Tiny Devices

Discussions: Why TinyML

- What kinds of machine learning models makes sense on tiny embedded devices
- What are the potential challenges

TinyML System Challenges

Limited Amount of Resources



A Typical Tiny Device

- Extremely limited memory resources
- Limited instruction set support(e.g. no floating point units)

Limited System Support

- No standard OS support: no files, dlls
- No virtual memory and malloc
- Limited programming languages(usually C)

Discussions

- How would these challenges impact ML applications
- What are possible ways to resolve these challenges

Model Quantization

Why Quantization

- Convert floating point operations to integer operations(usually int8)
- Reduce weight size
- Make use of integer arithmetic

Symmetric Quantized Representation

Use a pair (d, s) to represent the value

$$x = d * s$$

Original floating point number

Integer value

Scale

Quantized Arithmetic

Effective value bits

Quantize(s): convert to Integer $x \to (\text{round}(\text{clip}(x/s, 2^b - 1)), s)$

Requantize(s1, s2): convert between different scales $(d, s_1) \to (\text{round}(\text{clip}(d * s1/s2), 2^b - 1), s_2)$

Dequantize(s): convert back to floating point $(d, s) \to d * s$

Multiplications in Symmetric Quantization

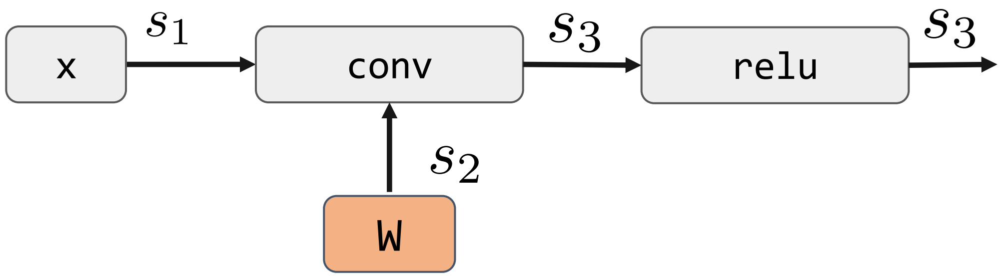
$$x_1 * x_2 = (d_1 * d_2) * (s_2 * s_1)$$

Integer value
usually need higher
amounts of bits to store

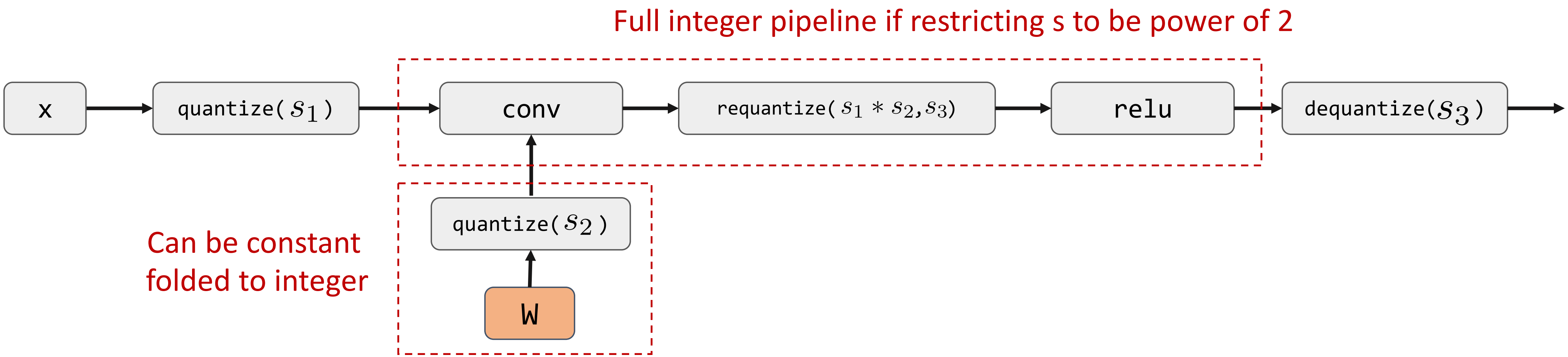
New scale

Representing Quantized Model

Attach the scale on the output of each layer



Convert to the integer representation



Discussions

- How can we decide the scale in each layer?

- How to handle re-quantize in a full integer setting

Calibration: Deciding the Scale of Each Layer

Quantize(s): $x \rightarrow (\text{round}(\text{clip}(x/s, 2^b - 1)), s)$

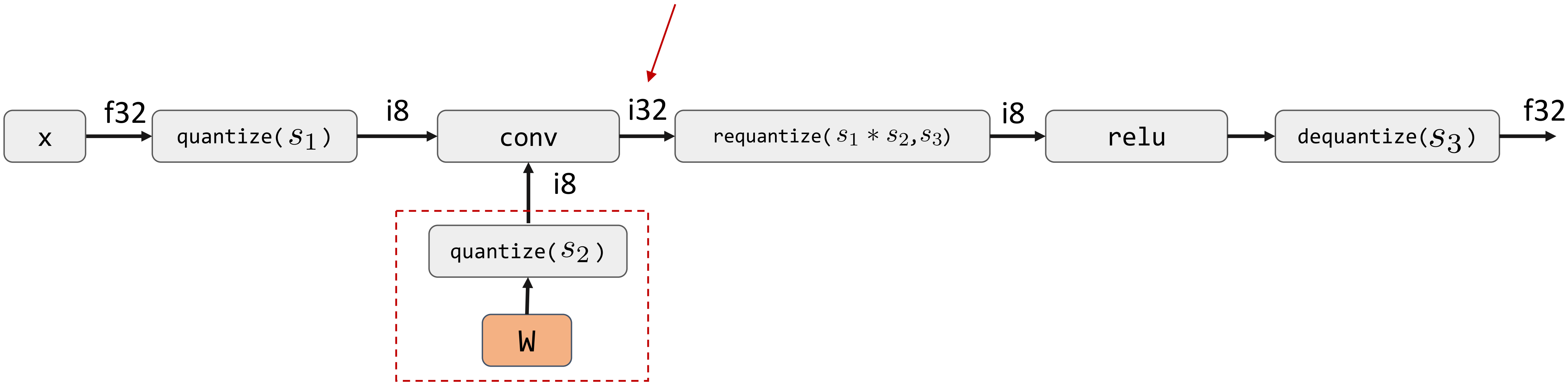
Two source of errors:

- Rounding error
- Clip by maximum number of bits

Compare and minimize difference between x and Dequantize(Quantize(x, s)) according to data distribution

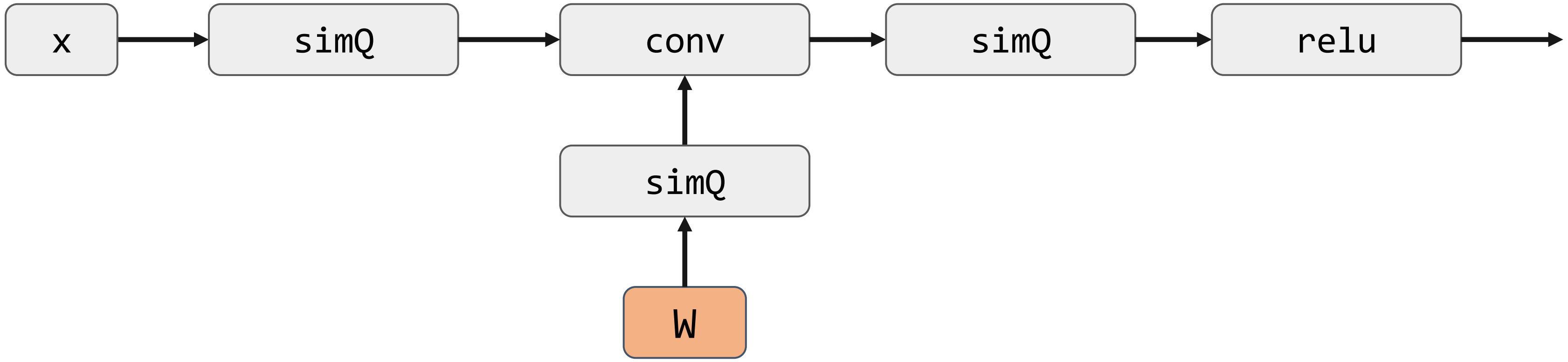
Mixed Precision in Integer Inference

User higher amounts of bits for accumulation



Quantization Aware Training

Fix a global scale, insert simulated quantization into the pipeline to simulate the error obtained due to quantization



simQ(x, s) = Dequantize(Quantize(x, s), s)

Discussions

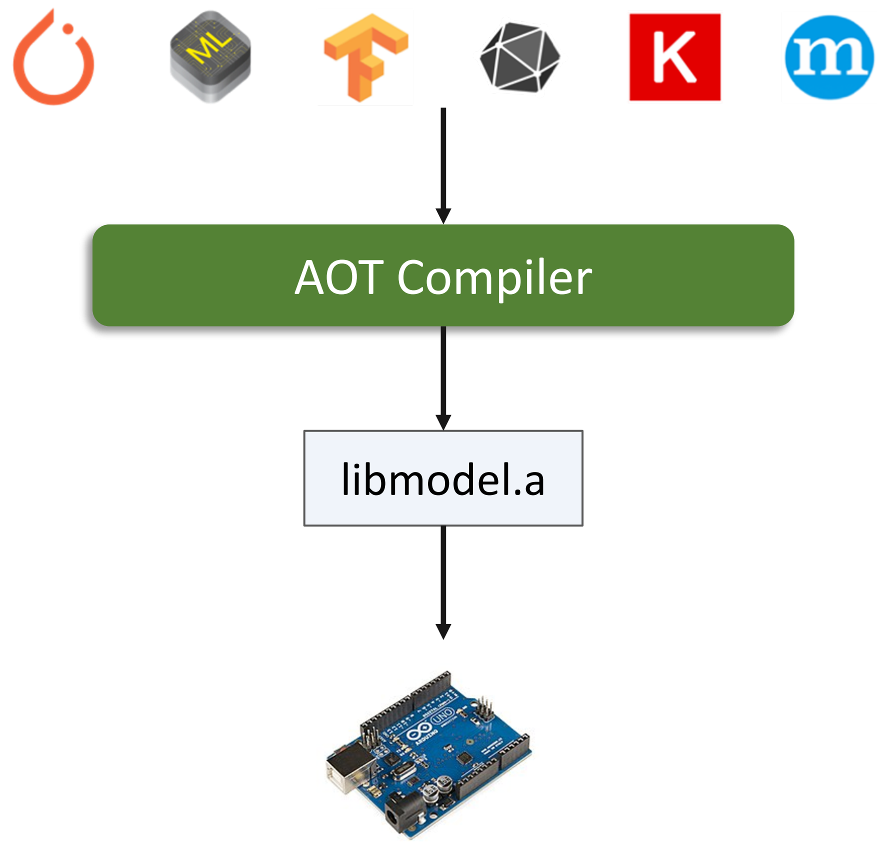
- What are other possible integer number representations other than the scale-based quantization?
- How to implement them effectively in embedded settings?
- How to support other neural network operators in full integer setting?

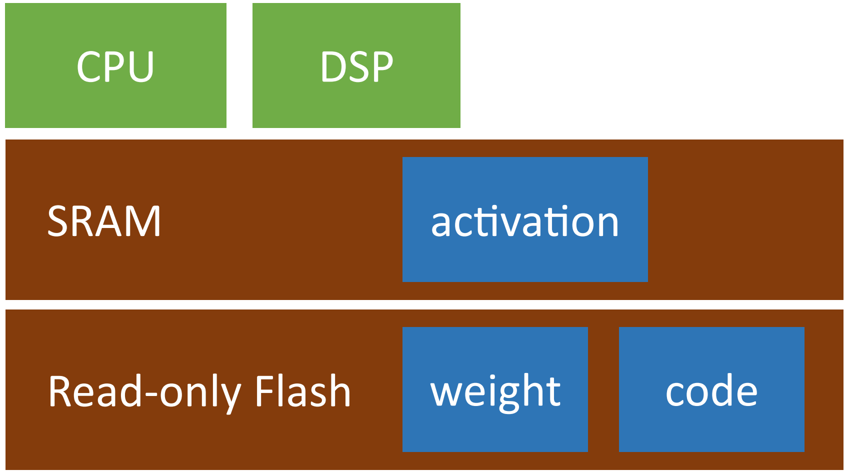
Beyond 8bit Integer

- Change accumulator bits (use i16 instead of i32)
- Smaller amount of input bits (use i4, i1)

Direct Model Compilation Approach

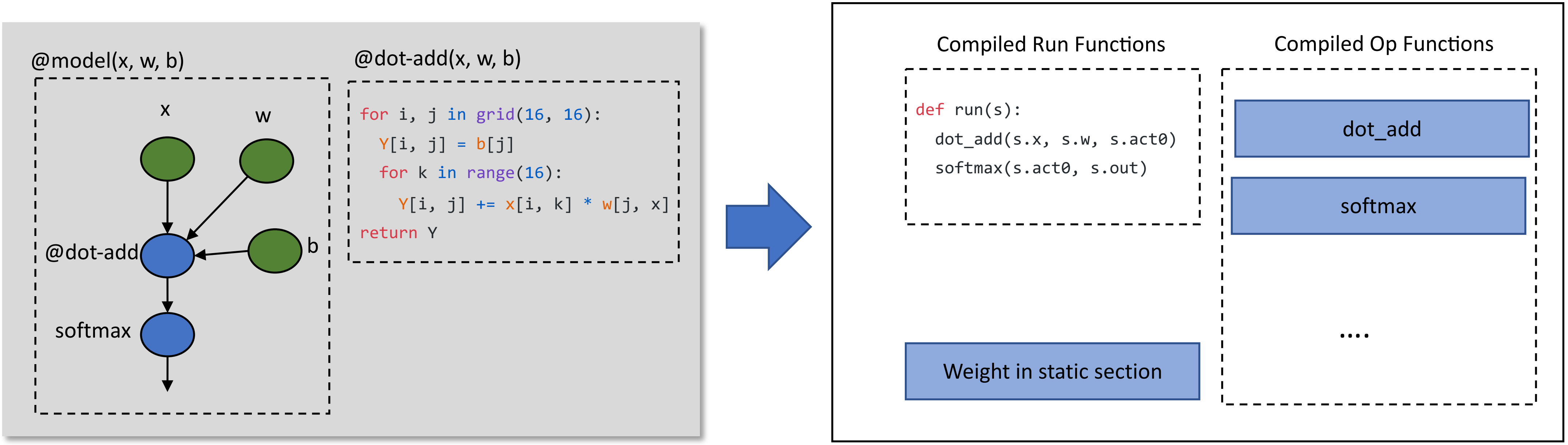
Ahead of Time Compiler based Approach





- Store weight on flash
- Use SRAM to store intermediate activations

AOT Compiler

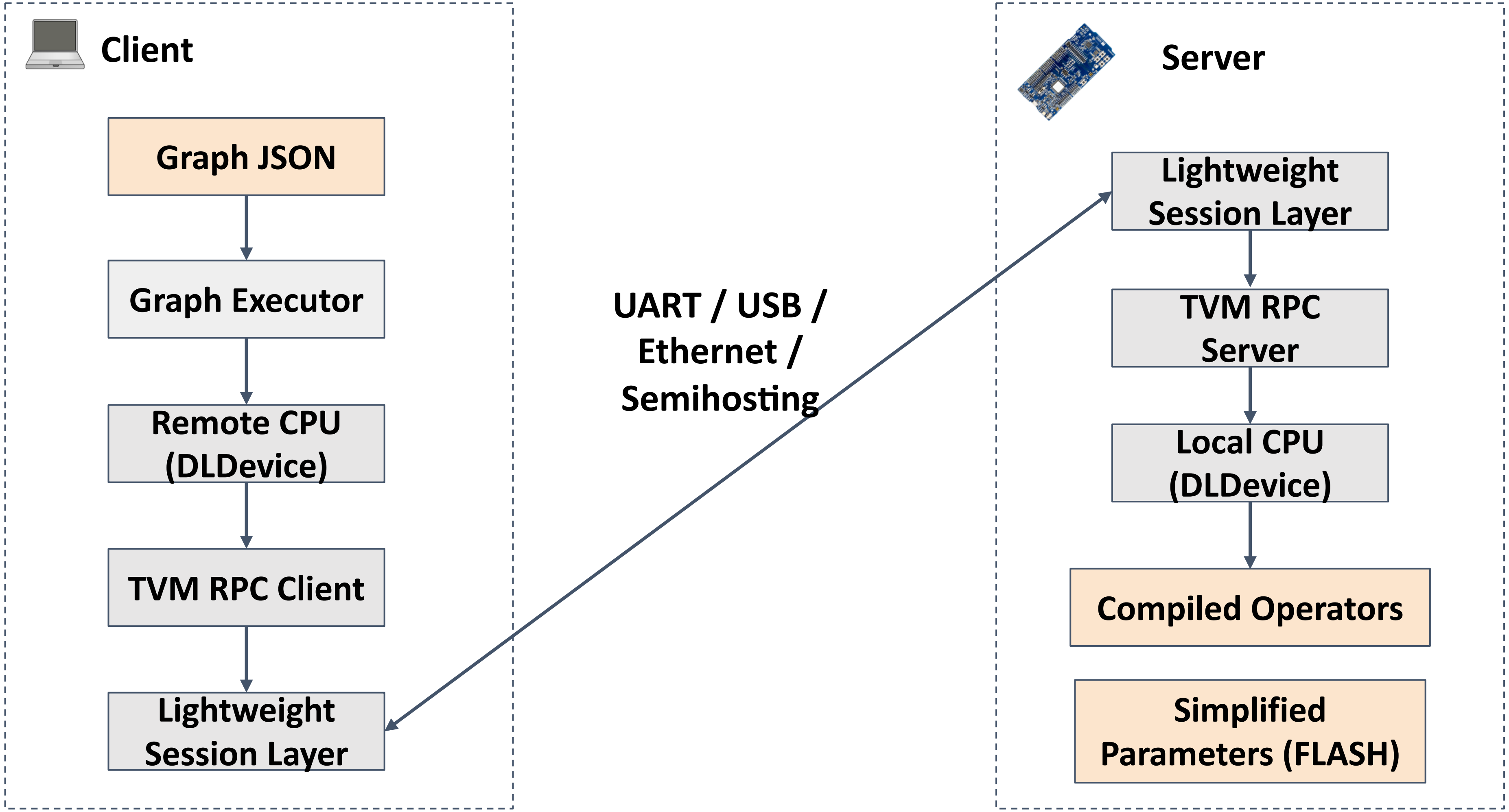


Runtime Execution

Discussion

- What are the complications when building a AOT compiler
- How to handle memory allocations

Solving Automation Infra Challenges: uTVM



Summary

- Tiny ML brings new challenges
- Algorithm approaches to model pruning and quantization
- System approaches to reduce the memory footprint

Logistics

Informal mid-term check-in (required, deadline April 18)

- Come to one of the office hours to talk about your current progress in the project

- Alternative: send a short email note about your current progress

Guest Lecture next week, separate zoom link, see piazza on thursday