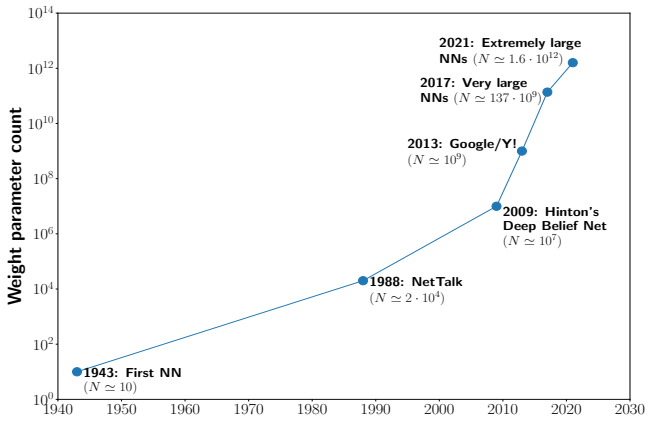
LeanAI: Dynamic Precision Training on the Edge

IRISA, TARAN
LS2N, OGRE
LIP, OKHAM

Context and objectives

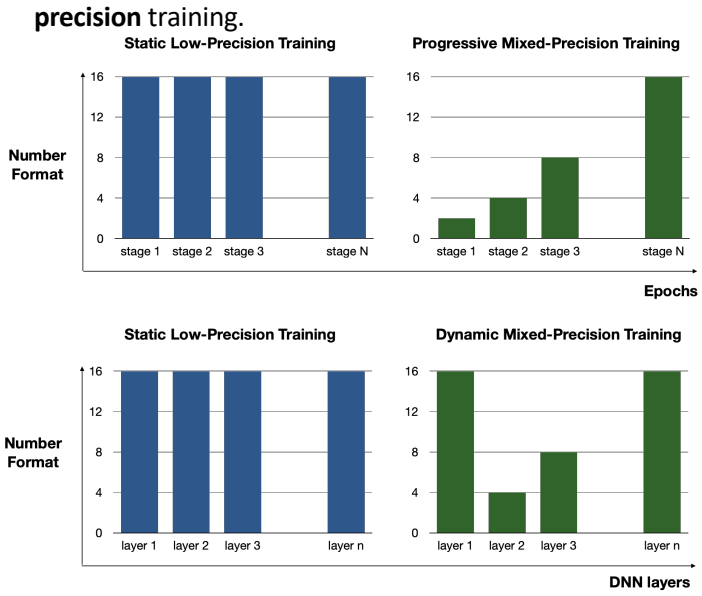
State-of-the-art deep learning models are growing fast.

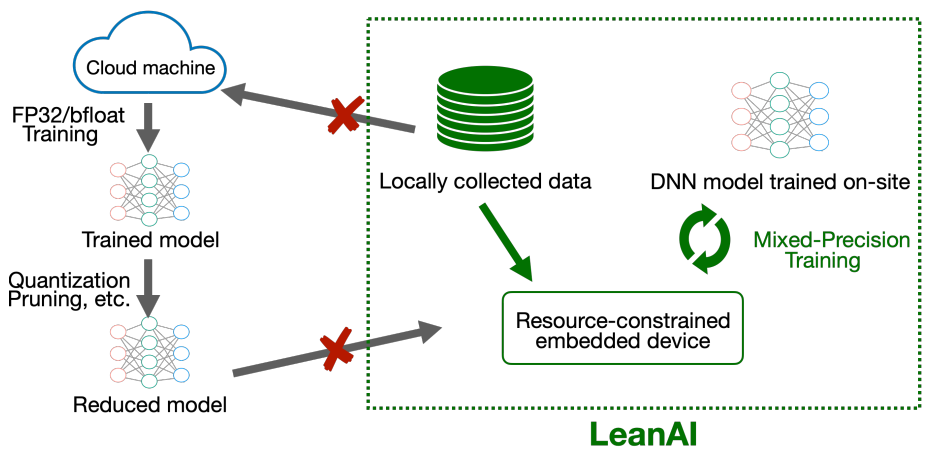


Context: need for **learning acceleration** mechanisms in both ***cloud*** (for large-scale models) and ***on-site*** settings (e.g. autonomous driving, healthcare, privacy).

Our approach

Accelerate training approaches jointly at the **algorithmic and arithmetic** levels. Explore **progressive and dynamic precision** training.





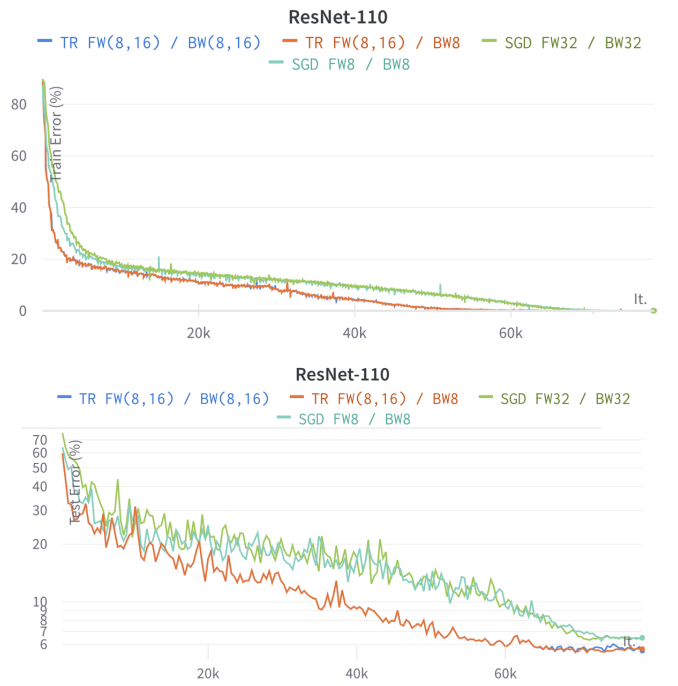
Objective: co-design a set of **optimization algorithms and tools** for deployment/synthesis on existing and custom HW.

Provide proof of concept **HW implementations** of the ideas developed to showcase potential impact of our results.

A dynamic precision trust region training algorithm

We are working on a trust region-based stochastic gradient descent algorithm that dynamically changes precision during training.

Initial results: ResNet-110 network trained on CIFAR-10 dataset; mixed precision training regime of 8-bit and 16-bit floating-point arithmetic.

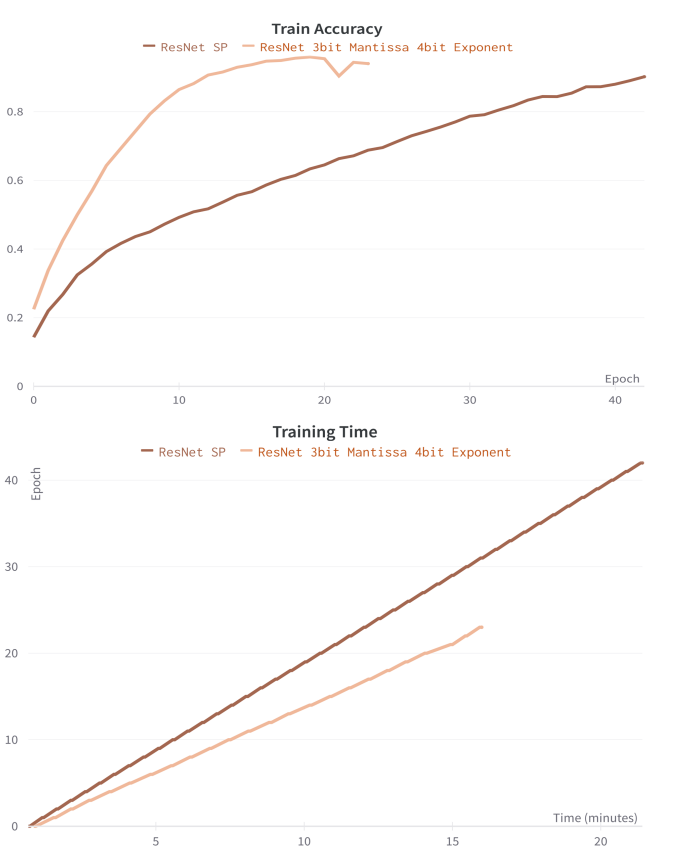


Custom precision activation function code generation

Explore precision switching for computing activation functions during the forward and backward pass and generate custom accelerated CUDA implementations.

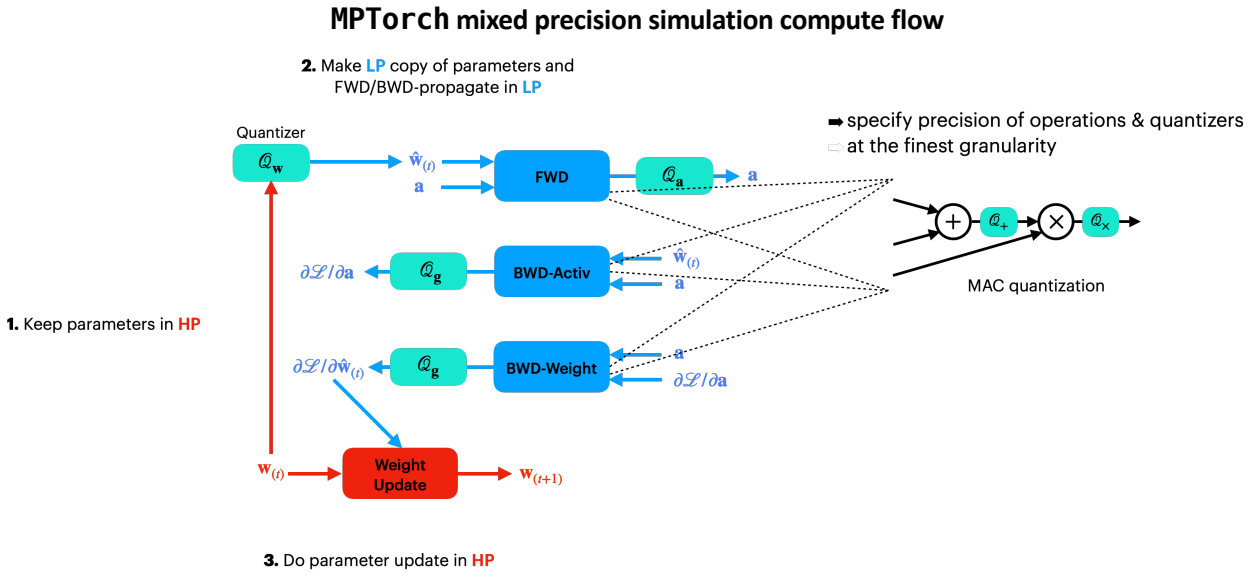
Initial results: activation function code generator based on Sollya and MetaLibm libraries; tests on GeLU-type activation functions and comparison with PyTorch implementation.

- need to optimize execution time



MPTorch: a custom precision training simulation framework

Library/simulation support for low/mixed precision training is limited. We created a PyTorch-based extension for low/mixed precision training (& inference) simulation.



Supports floating-point and fixed-point arithmetic. Planned support for various number formats and rounding modes.