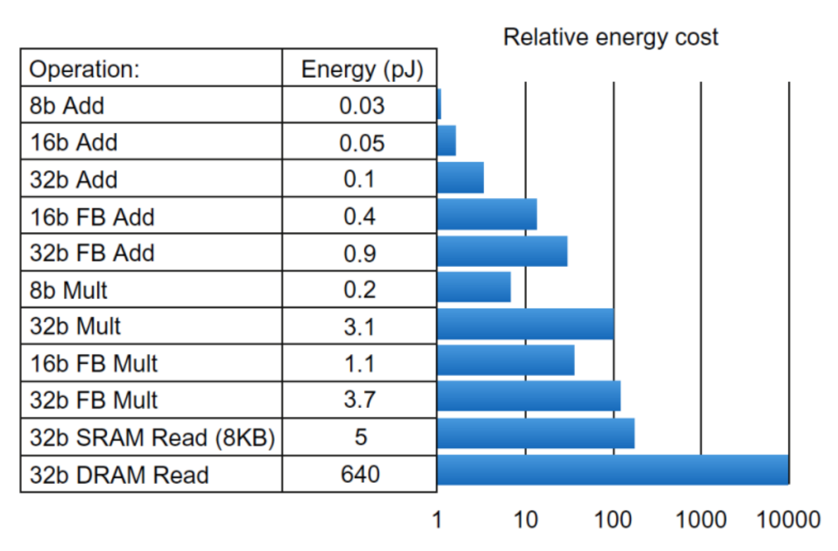
Dynamic Precision Training of Deep Neural Networks

Paul Estano Silviu I. Filip Elisa Riccietti

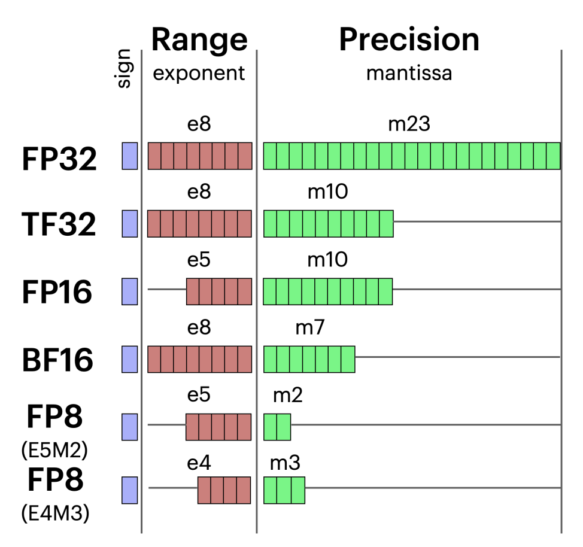
Universite de Rennes 1, IRISA, INRIA

Mixed Precision Learning

- **Objective**: make DNN training more efficient & scalable (for resource-constrained targets)
- Numerical values are representable using various formats (e.g., 32-bit, 16-bit, and 8-bit floating-point arithmetic)


Relative energy cost

Operation:	Energy (pJ)
8b Add	0.03
16b Add	0.05
32b Add	0.1
16b FB Add	0.4
32b FB Add	0.9
8b Mult	0.2
32b Mult	3.1
16b FB Mult	1.1
32b FB Mult	3.7
32b SRAM Read (8KB)	5
32b DRAM Read	640



[Horowitz et al., 2014]

- A small number format significantly reduces the energy cost
- Every function $f(x)$ is accessible up to a precision ϵ, through its low/mixed-precision approximation $\hat{f}(x,\epsilon)$:

$$|\hat{f}(x,\epsilon) - f(x)| \leq \epsilon$$

Mixed Precision Stochastic Gradient Descent (MP-SGD)

Extend Stochastic Gradient Descent by imposing a constraint on the precision.

$$\forall k \in \mathbb{N}^* : k \leq N,\ \epsilon_k \leq \eta \frac{\alpha_k}{2} \|\hat{g}(x_k, \epsilon_k)\|^2$$

$$x_{k+1} = x_k - \alpha_k \hat{g}(x_k, \epsilon_k)$$

With $\hat{g}(x_k, \epsilon_k)$ *stochastic gradient* up to a precision ϵ_k and $\eta < 1$

Theorem

With f possibly non-convex and L-smooth, MP-SGD converges to a stationary point. After N iterations we have:

$$\frac{1}{L}\mathbb{E}[\|\nabla f(x_R)\|^2] \leqslant O(\frac{1}{\sqrt{N}})$$

where the expectation is taken with respect to R

Pros

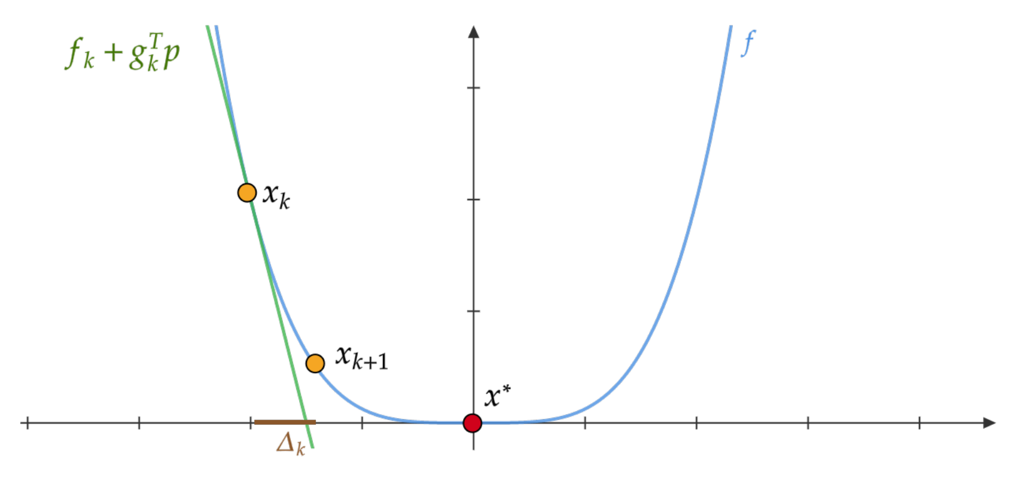
- Theoretical convergence guarantee
- Seamless (no need to modify the update)
- Same complexity as SGD

Cons

- Requires strong assumptions on the step-size α_k
- Requires knowledge of L
- There are potentially better alternatives to SGD in practice
- Slow convergence rate

Neural Networks Training in Trust-able Precision

- Trusts and optimizes a simple model m, $m_k(p) = f_k + g_k^T p$ within a ball
- The step-size is automatically chosen
- Extend [Gratton et al., 2019]
- Uses a stochastic setting and a 1st order model
- Same complexity as SGD



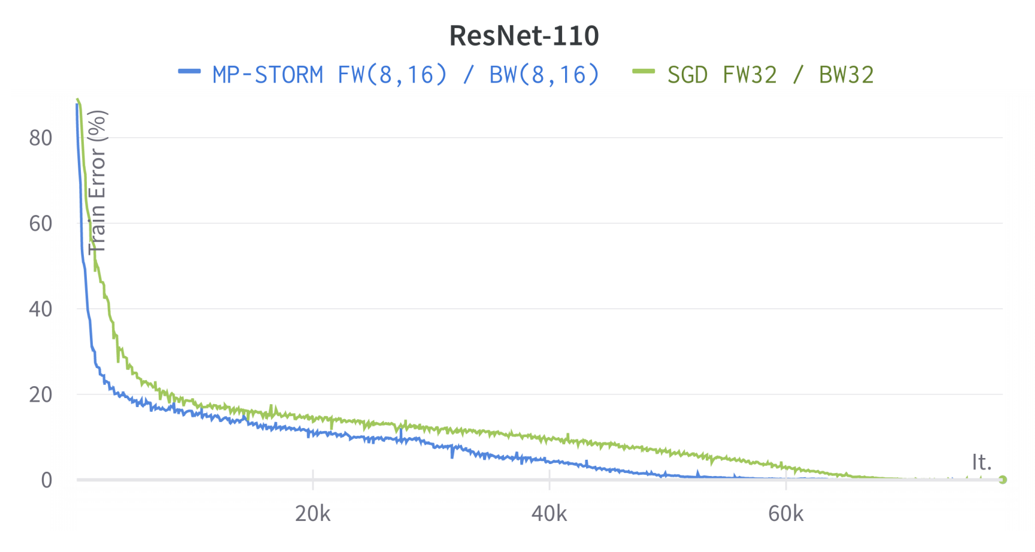
MP-STORM Algorithm

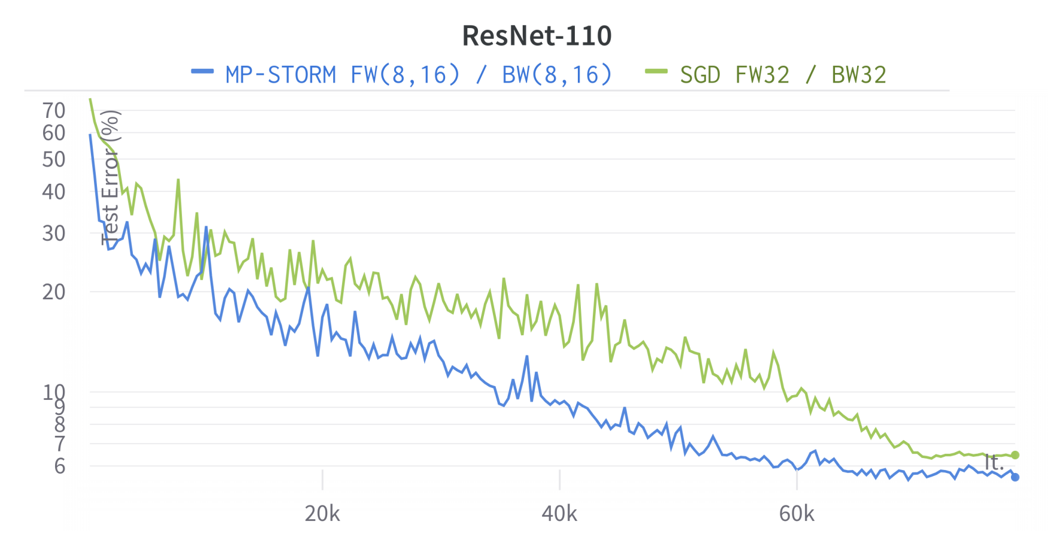
```
for $k = 1, 2, \ldots, N$ do
    1. Compute the step $p_k$
    2. Set the precision depending on $\rho_k$
    3. Update $\rho_k$ with the new precision if required
    4. Accept or reject the step depending on $\rho_k$
    5. Update the trust-radius $\Delta_k$ depending on $\rho_k$
end for
```

$$\rho_k = \frac{\overbrace{f(x_k) - f(x_k + p_k)}^{\text{actual reduction}}}{\underbrace{m_k(0) - m_k(p_k)}_{\text{predicted reduction}}}$$

CIFAR-10 Experiments

Encouraging results on CIFAR-10, competitive with SOTA FP32 methods.





Intra-Layer Mixed Precision for Neural Networks Inference Acceleration

We derive a bound on the error propagated in the layers of a multi-layer perceptron due to mixed precision computation.
This bound depends on the condition numbers of the weight matrices and the activation functions.
We validate this bound through experimental results. In practice, we choose the precision based on the values of the condition numbers.

Layer-wise Error Analysis

The error committed at layer ℓ is bounded by ϵ_ℓ such that:

$$\epsilon_\ell^h = \kappa_{\phi_\ell}(v_\ell) \circ \kappa_{v_\ell} \circ (\epsilon_\ell^W + \bar{\epsilon}_{\ell-1}^h) + \epsilon_\ell^\phi,$$

where:

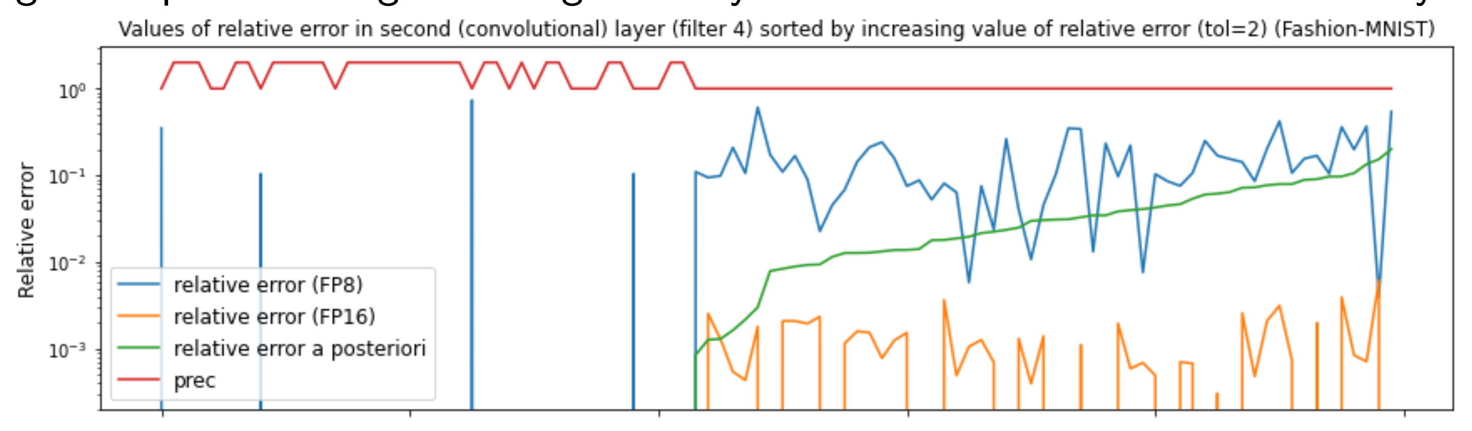
- κ_{ϕ_ℓ} condition number of the activation function
- κ_{v_ℓ} condition number of the weights/inputs vector product
- ϵ_ℓ^h, ϵ_ℓ^W, ϵ_ℓ^ϕ, $\bar{\epsilon}_{\ell-1}^h$ various errors committed at layers ℓ and $\ell - 1$

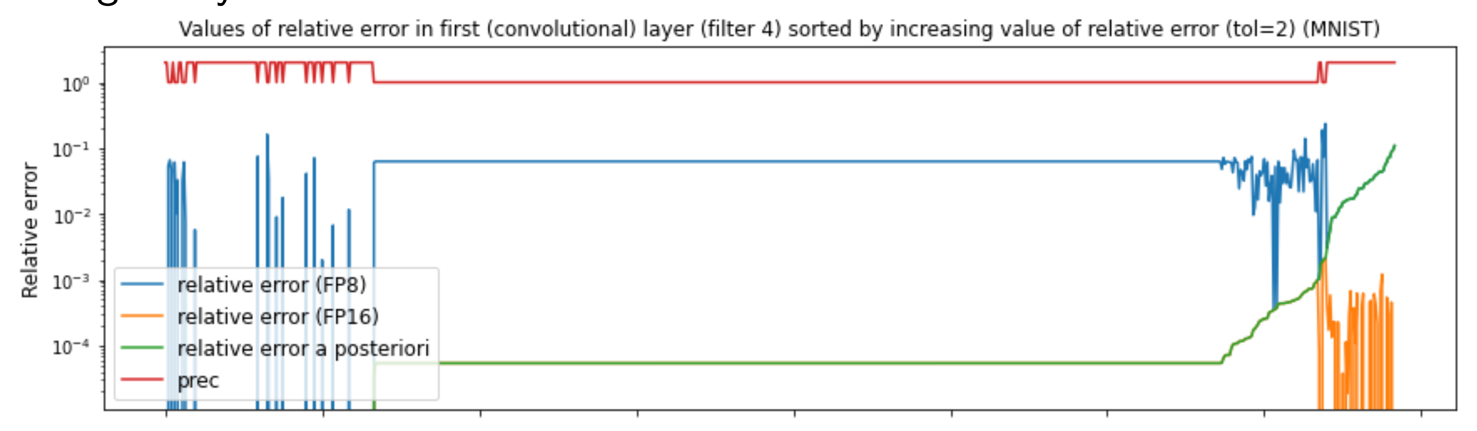
A Posteriori Mixed Precision Linear Layer Computation

```
1. Compute the layer output in full precision
2. Compute condition numbers $\kappa_{\phi_\ell}$ & $\kappa_{v_\ell}$ and their product in full precision
for every element $\kappa_i$ in the vector $\kappa_{\phi_\ell} \circ \kappa_{v_\ell}$ do
    if $\kappa_i \leq$ tolerance then
        $\epsilon_{\ell,i}^h \leftarrow$ FP8
    else
        $\epsilon_{\ell,i}^h \leftarrow$ FP16
    end if
end for
3. Compute the layer using $\epsilon_\ell^h$ values as precision for every dot product in the layer
```

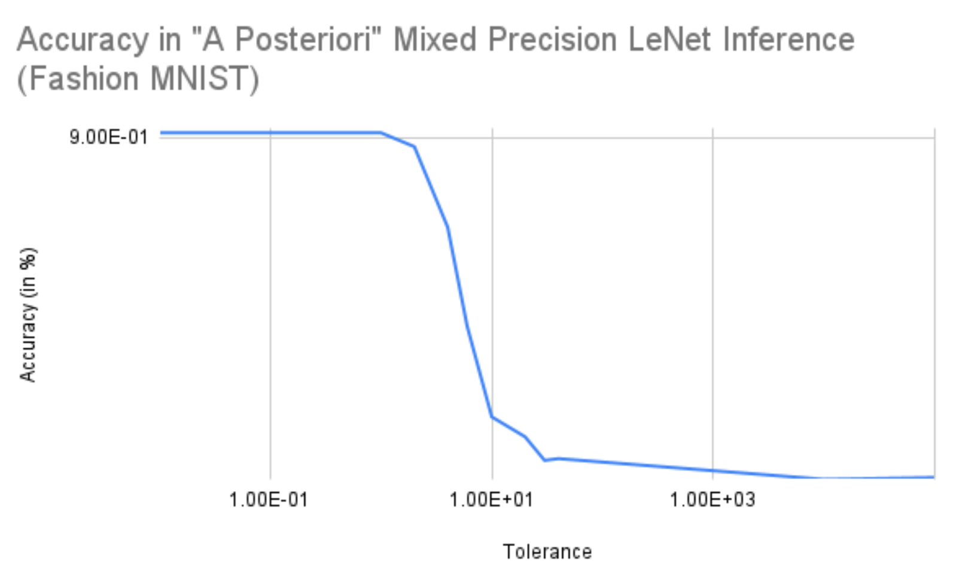
Experiments on MNIST and Fashion-MNIST with LeNet-5

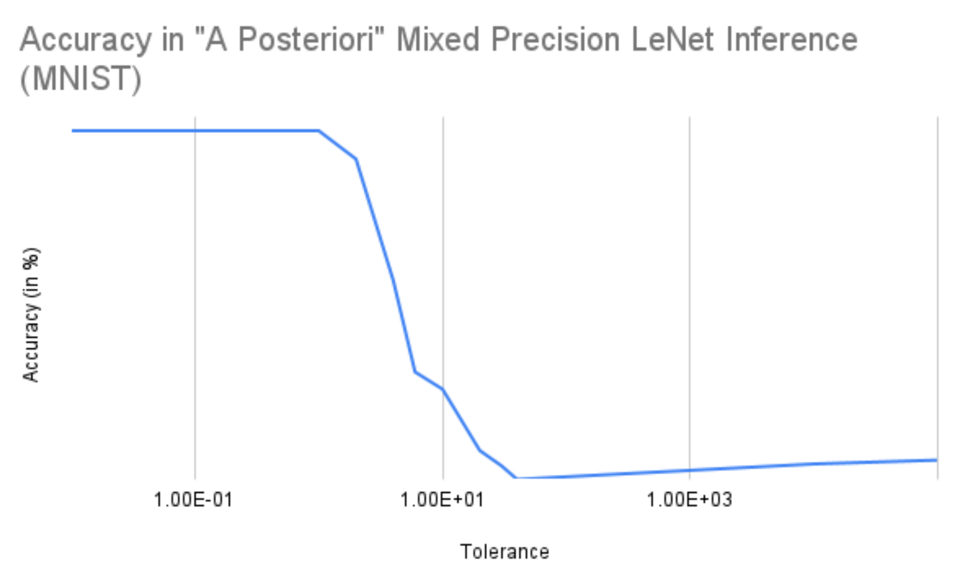
Selecting the precision using the a posteriori algorithm significantly reduces the relative error at each layer and globally.





The error (to a large extent) depends on the activation functions condition numbers $\Rightarrow$ precision of a given network can be chosen independently from the data.


Accuracy in "A Posteriori" Mixed Precision LeNet Inference (Fashion MNIST)
9.00E-01
Accuracy (in %)
1.00E-01
1.00E+01
1.00E+03
Tolerance


Accuracy in "A Posteriori" Mixed Precision LeNet Inference (MNIST)
Accuracy (in %)
1.00E-01
1.00E+01
1.00E+03
Tolerance