GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

Ahead-of-time (AOT) compilation of Tensorflow models for deployment

Authors: Bogdan Wiederspan, Marcel Rieger, Peter Schleper

The constraints of models in CMS workflow

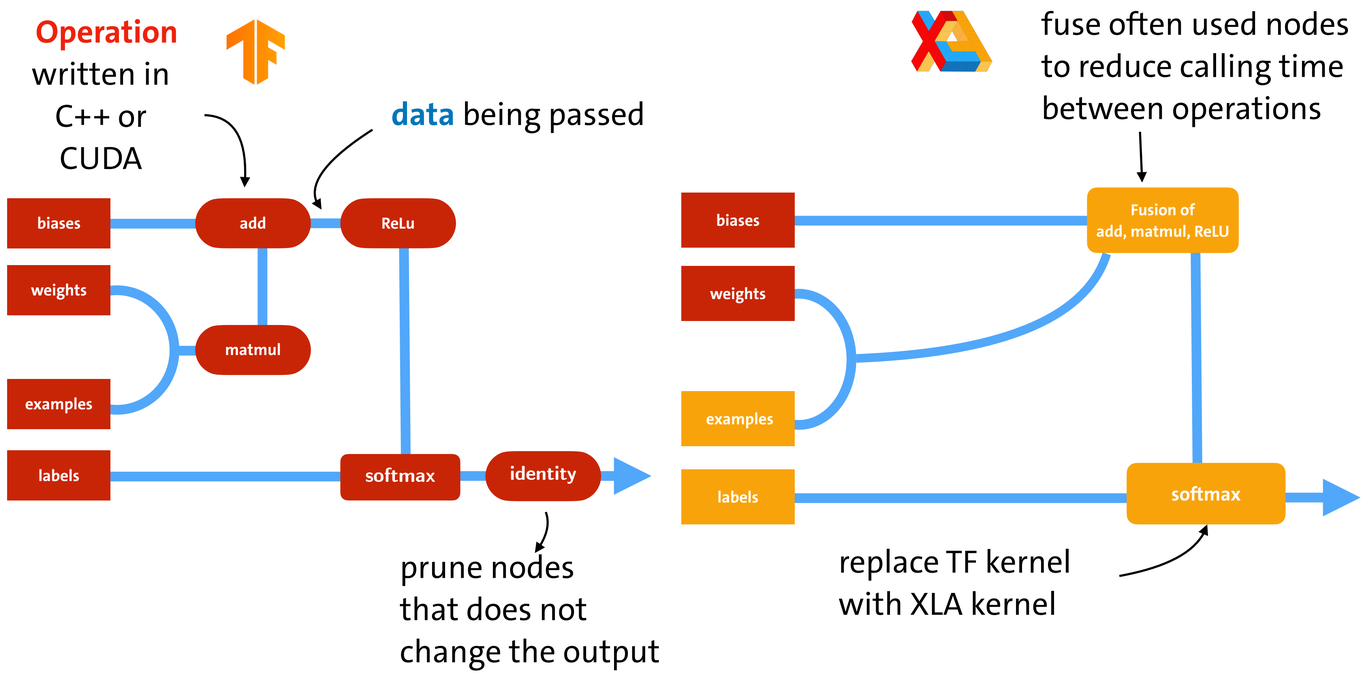
In the past years, various machine learning applications were integrated in central CMS workflows, leading to great improvements in reconstruction and object identification efficiencies. However, the continuation of successful deployments might be limited in the future due to memory and processing time constraints of more advanced models evaluated on central infrastructure. Currently, 12 models are deployed, each taking ~**100 MB of RAM**. The CMS workflow runs multithreaded, resulting in a upper limit of approx **2 GB RAM per CPU core.**

Accelerated Linear Algebra (XLA)

Computational graph represent the machine learning model and consist of **kernels** and **edges.** XLA enables model specific optimizations on graph or hardware level.

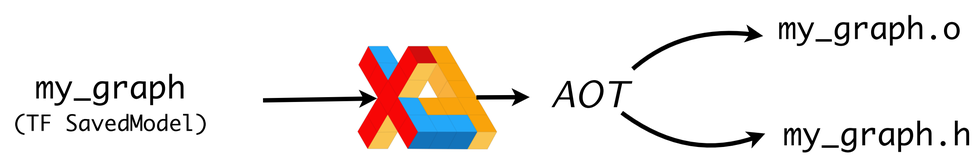
On graph level XLA replaces TF kernels, buffers are allocated, common subexpressions are reused and unused ops are pruned.

On hardware level architecture specific optimisation are done.



Ahead-of-time (AOT)

AOT compile of a graph means to compile it statically into **self-contained libraries (header-object pair).**
The models becomes a series of C++ compute kernels.



Advantages

- drastic reduced in memory footprint for inference
- independence of TensorFlow Framework
- easy Multi-threading behaviour

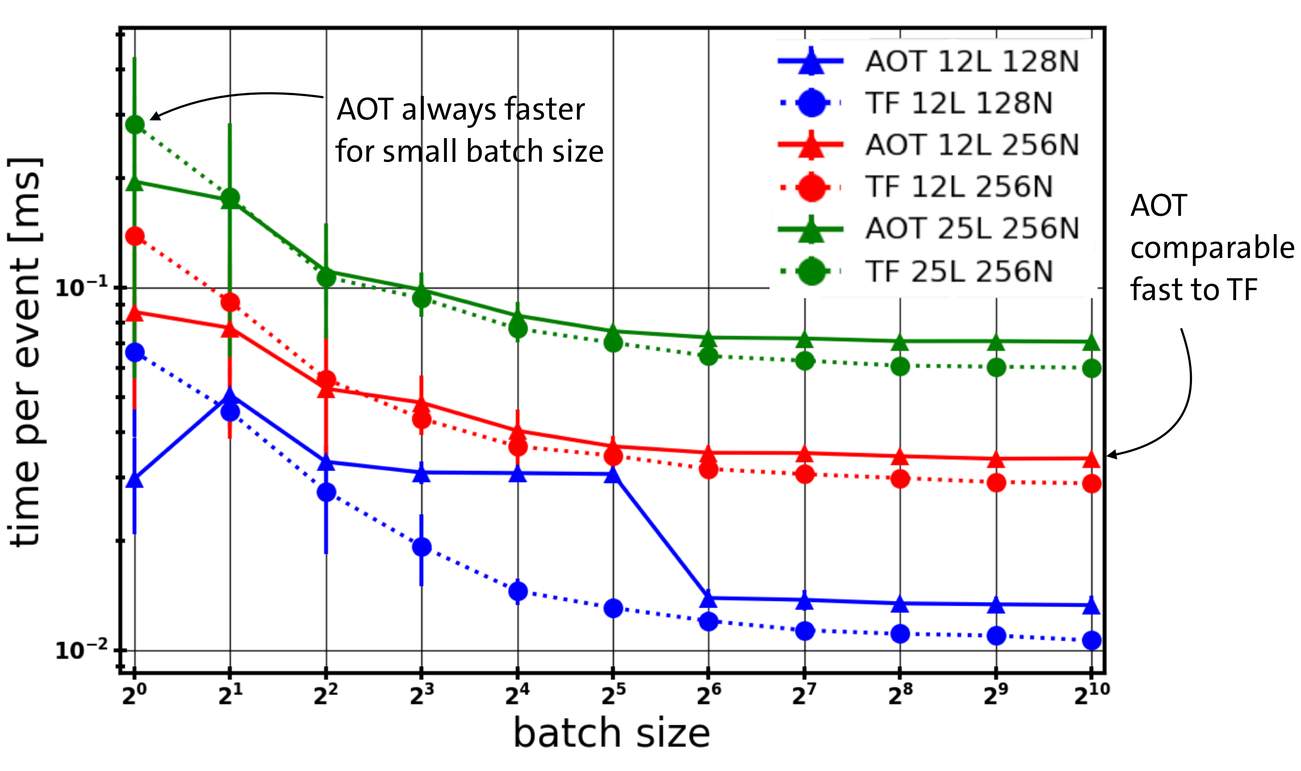
Disadvantages

- reservation of buffers require fixed memory layout at compile time
- dynamic batching needs to be emulated (padding/ stitching)
- existence of TF kernel conversion to XLA

CPU Performance

The model is a simple feed-forward model with batch-normalization with up to 25 layers (L) and 256 nodes (N).

Only default XLA optimization are applied, thus we see only **bare minimum**.



Memory Performance

Allocated memory is shown for a multithreading scenario. Measurement shows the sum of the runtime environment and the loaded models.

AOT has almost **no overhead** and the allocated memory is solely driven by the size of the weights, which is not true for TensorFlow models.