TMVA Deep Learning on GPU

L. Moneta, J. Niermann (CERN EP-SFT)

E4 project meeting, October 4th 2019
ROOT/TMVA

- ROOT is a software toolkit for data processing, storage and analysis in HEP
- Machine Learning tools are provided in the TMVA package (Toolkit for MultiVariate Analysis)
  - Set of algorithms for standard HEP usage
  - Used in LHC experiment production and in several analyses
- Key features
  - Facilitates HEP research, from detector to analysis
  - Easy to use, good performance
  - Long term support
  - Several new developments (e.g. deep learning)
- Development done in collaboration with CERN experiments and HEP community

http://root.cern.ch/tmva
Deep Learning in TMVA

- Deep Learning Library in ROOT/TMVA with support for
  - Dense (fully connected) layers
  - Convolutional layers
  - Recurrent layers

- Parallel implementation for both CPU and GPU
- Very efficient in training time
- Focus also for low latency in inference of trained models
DNN Training Performance

Training time — Dense networks (5 fully connected layers 200 nodes)

TMVA vs Keras (Tensorflow)
Convolutional Neural Network

- Available in ROOT/TMVA (since release 6.14)
- Supporting CPU parallelization
- GPU parallelisation and code optimisation is essential for fast training and inference

![Signal image](signal.png)

![Background image](background.png)

Convolutional + Pooling + Dense layers

Input 32x32 images

**Background rejection versus Signal efficiency**

- MVA Method:
  - CNN CPU
  - PyKeras
  - DNN CPU
ROOT CNN

- Original implementation based on kernels developed for convolution operations
- Kernel working at single event and not at batch level
- No performance improvements increasing batch size
Work on a new implementation using NVIDIA cuDNN library

- adapt ROOT/TMVA to use directly cuDNN functions for convolution operators
  - *(work performed with J. Niermann, OpenLab summer student, 2019)*

- re-designed code to use a new C++ Tensor class (RTensor)
- work almost completed, will be integrated for next ROOT release (6.20)
  - further optimisations still possible
- next step: implement also recurrent and LSTM layers
cuDNN Training Performances

- Using images data of 32x32 with 4 convolutional layers

![Graph showing the performance comparison between Keras (Tensorflow), TMVA (Cuda only), and TMVA (Cudnn) with varying batch sizes.](image)

- Larger = Better

- GPU card: RTX 2070, 8Gb
cuDNN Training Performances

- Using different images sizes (some toys data)

GPU card
RTX 2070, 8Gb
Next Steps

- Focus on inference of models
  - efficiency performance for single event evaluation
  - generic inference engine in ROOT/TMVA for trained models using ONNX format
- Support 3D convolutions
- cuDNN implementation for RNN / LSTM
Conclusions

- Optimisation of convolution operations for GPU using NVIDIA low level libraries (cuDNN) almost completed
  - excellent performances reached especially for small batch sizes/ small network architectures
- Big advantage for the community having optimized ML tools available in ROOT
  - can be deployed in reconstructions or real time applications
- Results will be presented at next CHEP2019 in Adelaide