Accelerating TMVA Deep Learning - Integration of the NVIDIA cuDNN Library

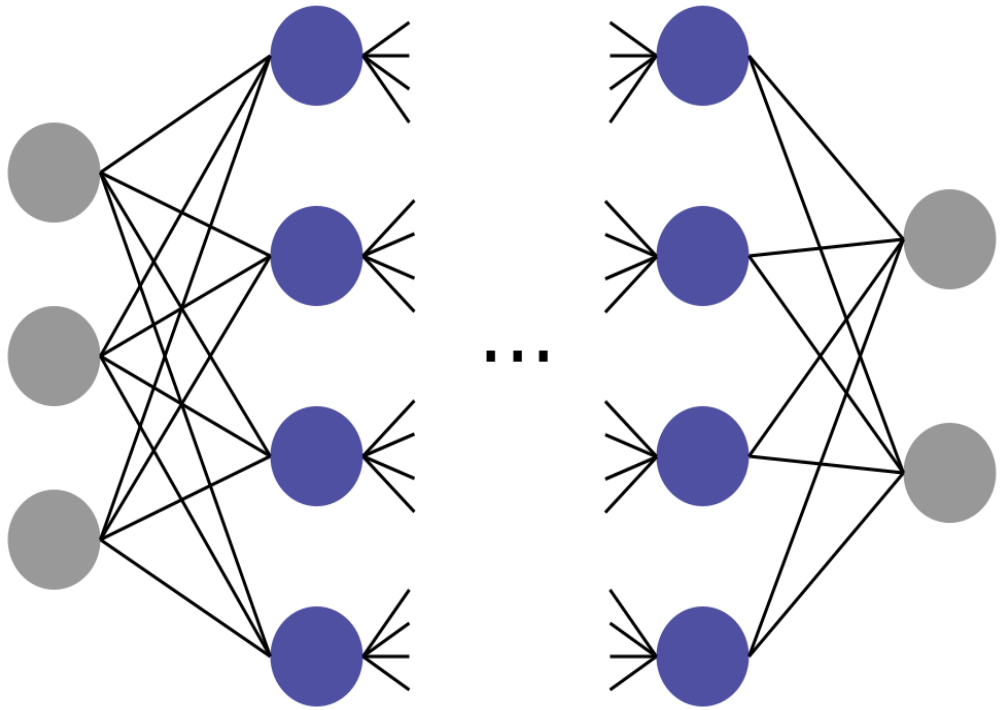
CERN openlab Lightning Talks

Joana Niermann

August 15, 2019

Adding Depth to Learning

Combination of multiple layers of neurons to learn complex concepts:



Ingredients for Training:

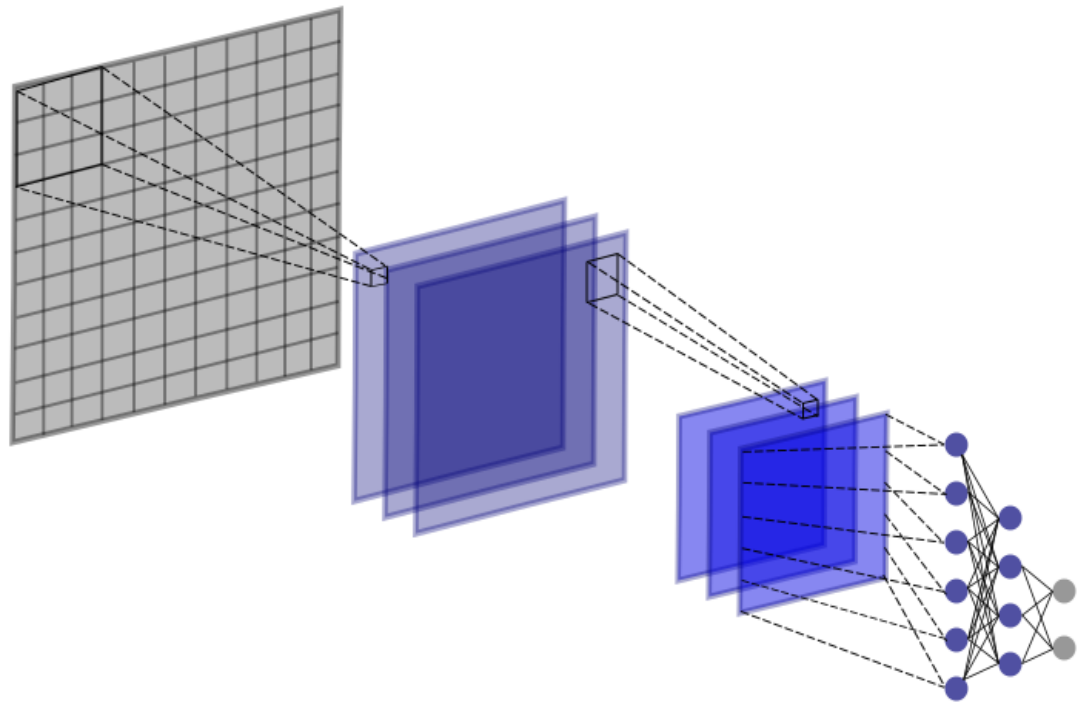
- *Weights, biases, activations, loss function, regularizations . . .*
- *Forward pass* gains current network response
- *Backward pass* calculates and propagates weight updates

Convolutional Neural Networks

Offer *convolution* operations with trainable filters

$\rightarrow$ feature extraction in images

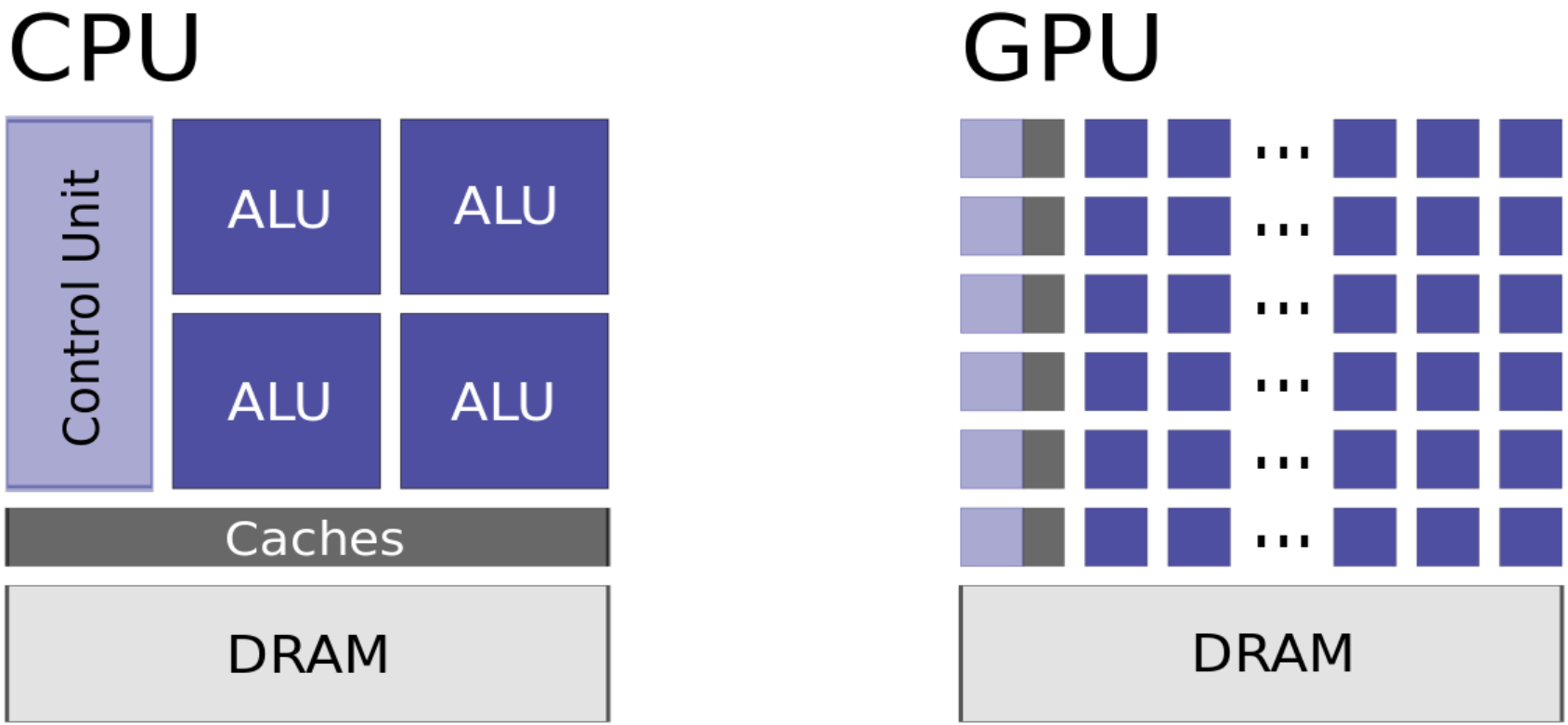
- *Pooling* yields dimensionality reduction
- Further layers for *batch normalization*,*reshaping*, *dense layers* etc.



Manifold usage in particle physics: E.g. “detector images“ for jet classification, object identification, tagging . . .

Deep Learning on GPUs

Neural network training and inference can be accelerated using parallel hardware: [1]



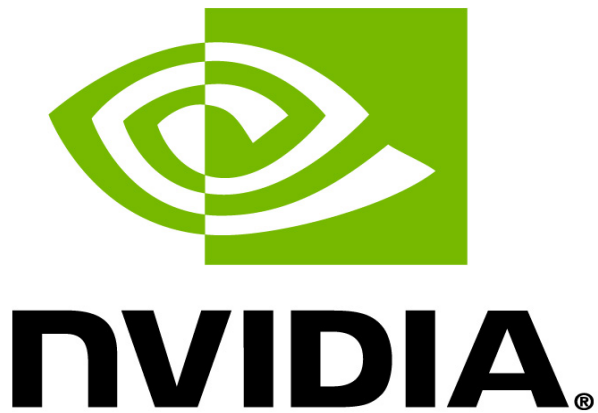
⇒ Time saved on training can be put into precision

[1] Image adapted from [4].

The NVIDIA cuDNN Library

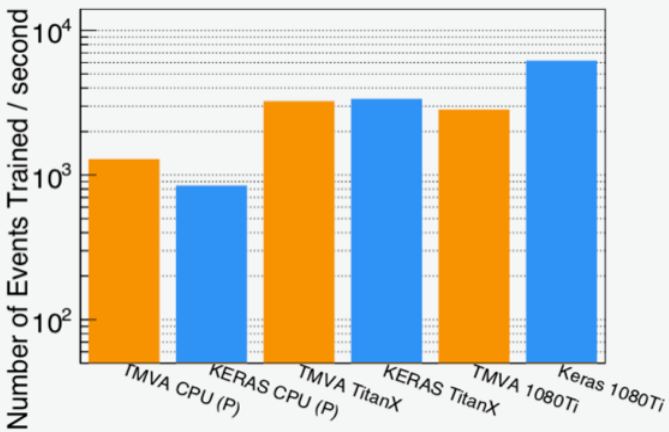
Exposes C style API for deep learning primitives

- Fast, memory efficient implementation
- Adapted to specialized hardware
- Easy to maintain

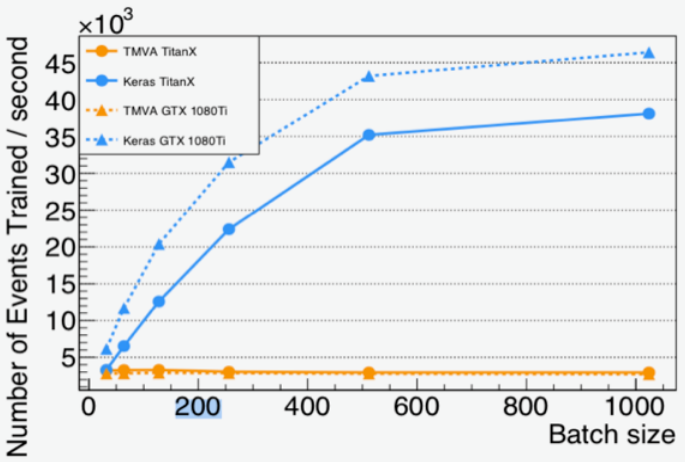


TMVA - A multivariate Analysis Project

- ROOT TMVA contains DNN, CNN and RNN implementations for CPU and CUDA enabled hardware
- Currently no batch training, but single input passes for CNN on GPU[2]



(a) Event throughput for different backends for batch size 32.



(b) Scaling for GPU implementations with varying batch size, from 32 to 1024 at increasing powers of 2.

⇒ Reimplement/Adapt the CNN and data handling classes to use cuDNN library and cuDNN tensor layout

[2] Image from [8].

Thank You!

References i

[1] Christopher M Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, NY, 2006. Softcover published in 2016.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[3] Dan Guest, Kyle Cranmer, and Daniel Whiteson. Deep Learning and Its Application to LHC Physics. *Annual Review of Nuclear and Particle Science*, 68(1):161–181, 2018.

[4] NVIDIA. NVIDIA CUDA C Programming Guide. *CUDA_C_Programming_Guide. pdf*, 2013. `https://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf`.

[5] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, et al. cuDNN: Efficient Primitives for Deep Learning. *CoRR*, abs/1410.0759, 2014.

[6] Rene Brun and Fons Rademakers. ROOT — An object oriented data analysis framework. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 389(1):81 – 86, 1997. `https://github.com/root-project/root`.

[7] A. Hoecker, P. Speckmayer, J. Stelzer, et al. TMVA - Toolkit for Multivariate Data Analysis. *arXiv e-prints*, page physics/0703039, Mar 2007.

[8] K. Albertsson et al. New Machine Learning Developments in ROOT/TMVA. CHEP 2018, Sofia, Bulgaria, 9-13 July 2018.