A OneAPI backend of **hls4ml** to speed up Neural Network inference on CPUs

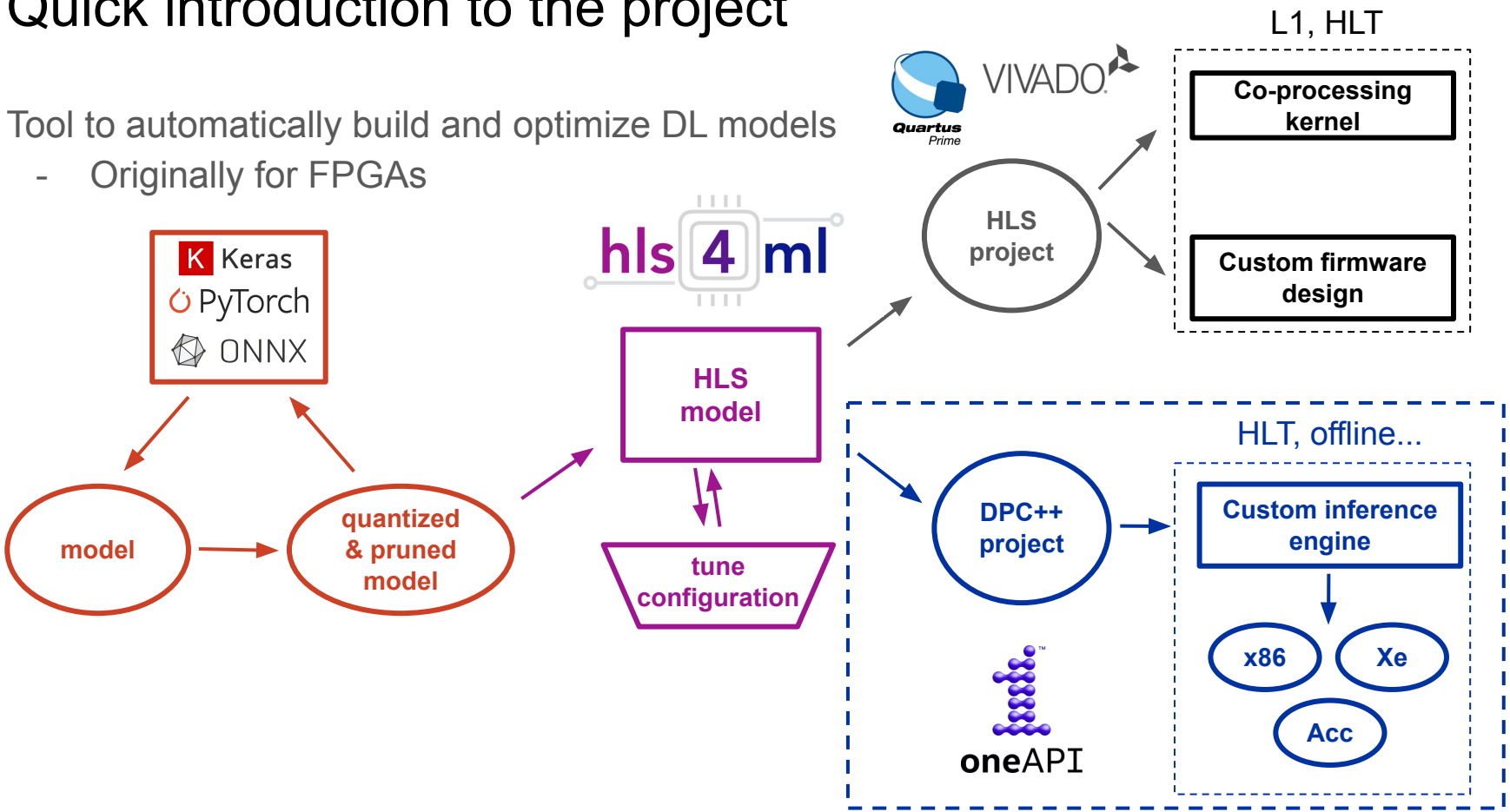
Vladimir Lončar
For the FastML team
fastmachinelearning.org

30.11.20
Southern Methodist University

Quick introduction to the project

Tool to automatically build and optimize DL models

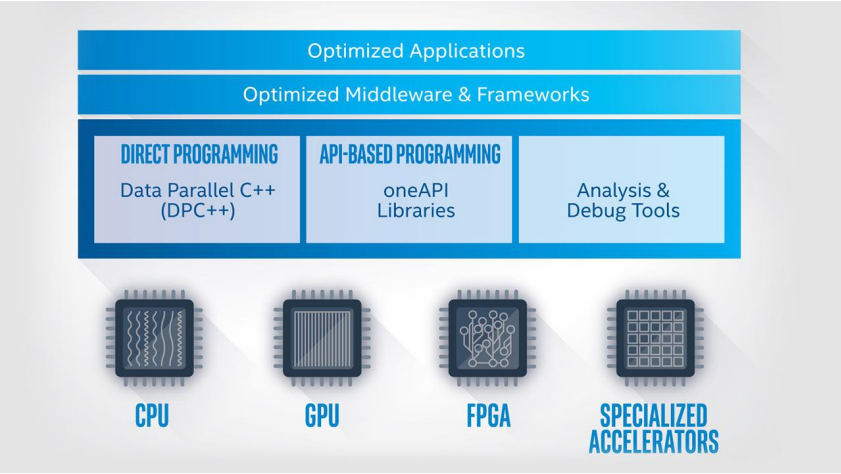
- Originally for FPGAs



Intel oneAPI

A core set of tools and libraries for building and deploying high-performance, data-centric applications across diverse architectures

- Data Parallel C++ (DPC++) language, an evolution of C++
- Allows code reuse across hardware targets - CPUs, GPUs, and FPGAs
- Permits custom tuning for individual accelerators



Intel oneAPI libraries

Intel oneAPI DPC++ Library (oneDPL)

Intel oneAPI Threading Building Blocks (oneTBB)

Intel oneAPI Math Kernel Library (oneMKL) 

Intel oneAPI Deep Neural Network Library (oneDNN) 

We are interested in these two

Intel oneAPI Data Analytics Library (oneDAL)

Intel oneAPI Collective Communications Library (oneCCL)

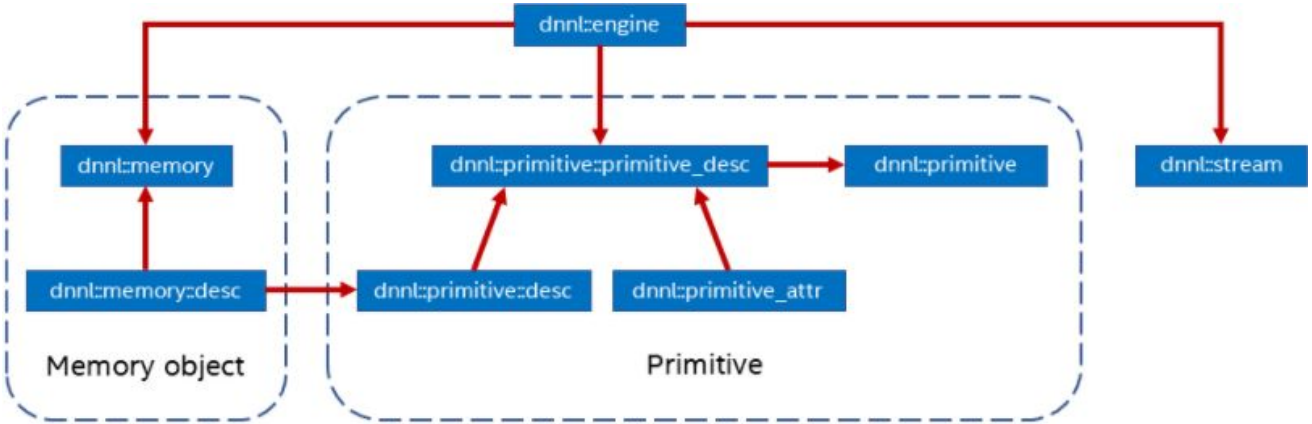
Intel oneAPI Video Processing Library (oneVPL)

...

OneDNN Programming Model

OneDNN challenges:

- Memory abstraction
- Mapping mathematical equations to oneDNN primitives
- Separation of network compilation and execution



OneAPI backend in **hls4ml**

Currently supported layers:

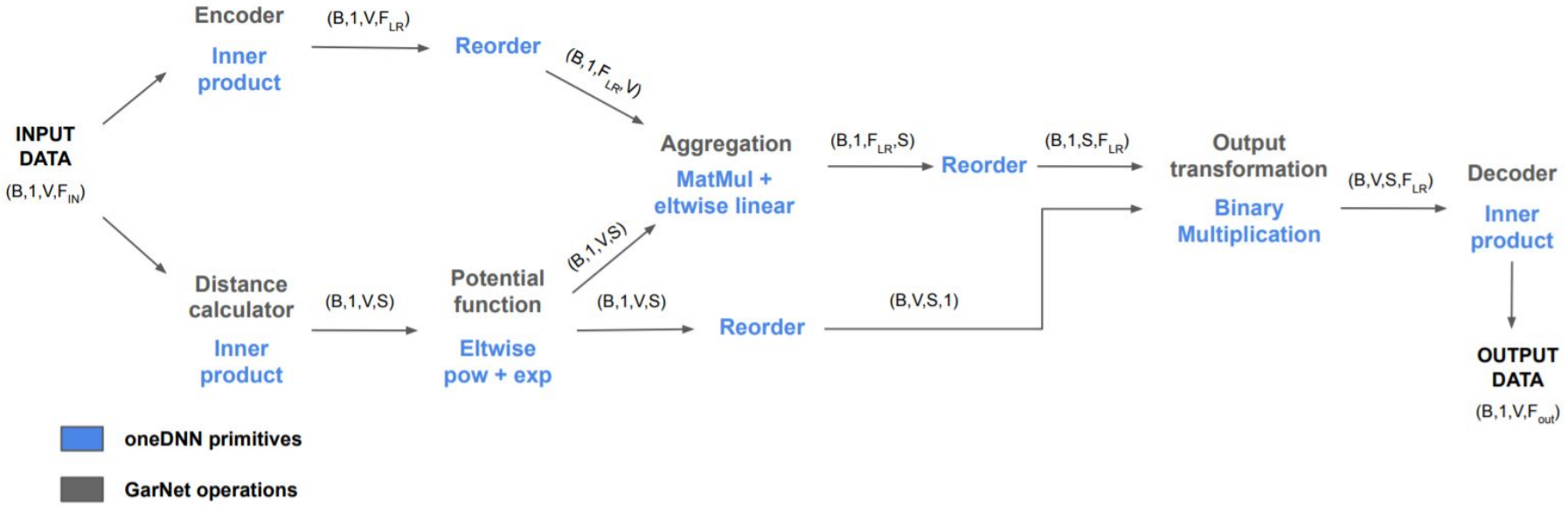
- Dense (inner product primitive)
- Activation functions such as ReLU, tanh, log, linear, exp, sqrt and many others (element-wise primitives)
- Softmax primitive
- Convolution primitive
- Pooling primitives (MaxPooling, AveragePooling)

All implemented using oneDNN primitives!

- Portable performance across multiple architectures

Custom Neural Networks using oneDNN

GarNet graph neural network - arxiv:2008.03601

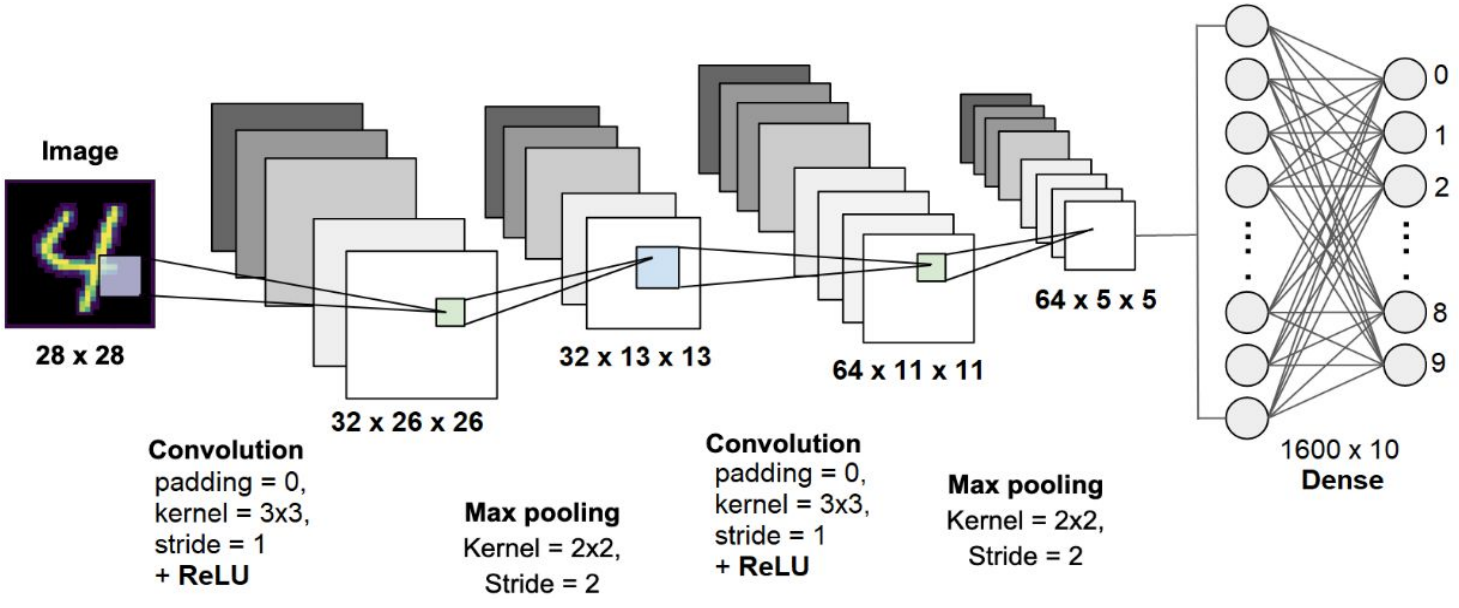


Performance evaluation

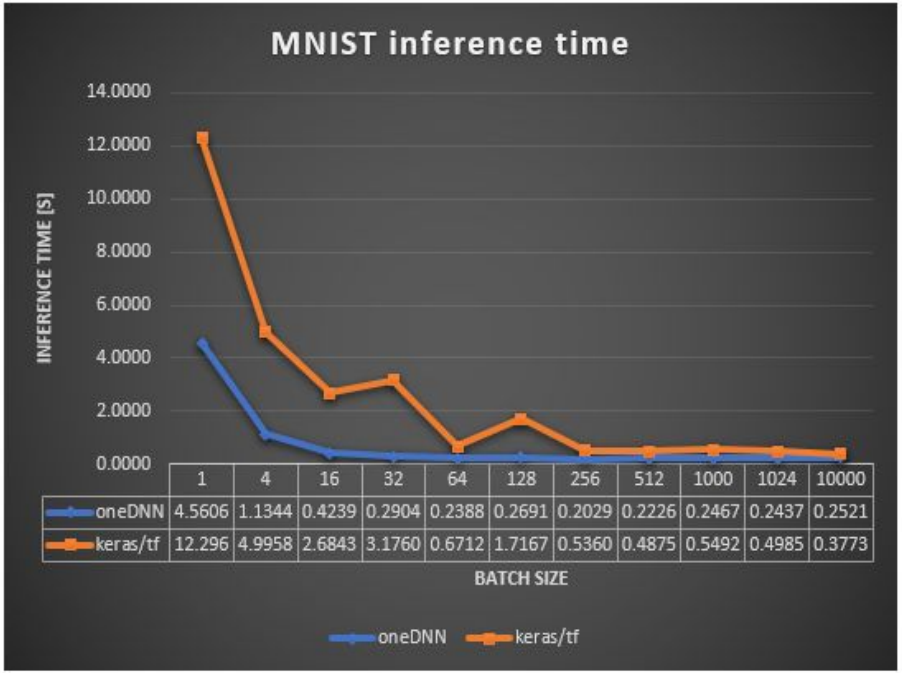
MNIST model

Tested in Intel DevCloud on Intel Xeon Gold 6128 processor

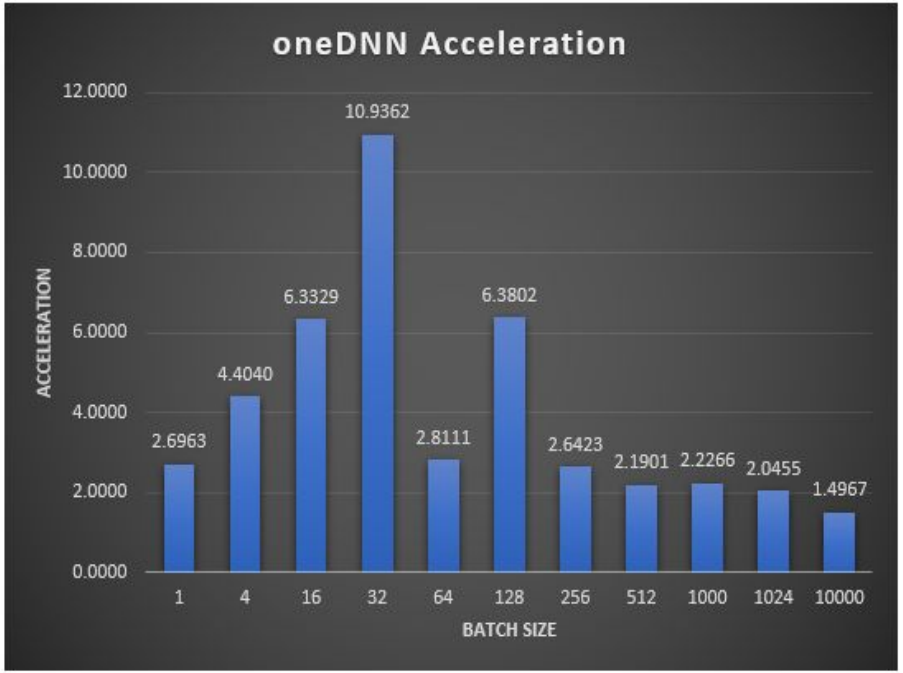
- 6 cores @ 3.40GHz



MNIST Digit Classification problem using oneDNN



	1	4	16	32	64	128	256	512	1000	1024	10000
oneDNN	4.5606	1.1344	0.4239	0.2904	0.2388	0.2691	0.2029	0.2226	0.2467	0.2437	0.2521
keras/tf	12.296	4.9958	2.6843	3.1760	0.6712	1.7167	0.5360	0.4875	0.5492	0.4985	0.3773



Possible paths of future development

Enabling quantization

- oneDNN supports 32-bit, 16-bit and 8-bit types in x86 hardware

Supporting more layer types

- Using the primitives available in oneDNN

Performance profiling and benchmarking

Deployment of the software on other Intel architectures

- Xe GPU architecture, Habana and (perhaps) FPGAs

Summary

OneAPI Programming model can speed up inference significantly on Intel CPUs

- Prototype hls4ml backend available as pull request

OneDNN Primitives can be used to design custom neural networks

- Prototype of GarNet model under development

The initial focus is on Intel CPUs, but we are not limited to them

- Happy to explore Intel's new architectures

Huge thanks to Marcin Świniarski @ Intel for creating this backend!