Building Heterogeneous Platforms for Online Learning with Dataflow Computing Design

Benoit Corda 1, Clement Farabet 1,2, Marco Scoffier 3, Yann LeCun 1

1. Courant Institute, New York University, 715 Broadway, New York
2. Electrical Engineering Department, Yale University, New Haven
3. Net-Scale Technologies, New Jersey

Summary

Instances of Machine Learning Models share a set of basic operations, in particular for neural networks, models are dominated by convolutions, dot-product and element-wise operations.

For general purpose CPUs, several machine learning libraries have been developed, which all provide efficient multi-dimensional arrays and modular description frameworks, this includes Lush [1], Torch [2] and Theano [3]. Essentially, all these libraries, and more generally all scientific packages (e.g. NumPy, Matlab, ... ) rely on compiled code to handle the most-repetitive operations, and provide a high-level, dynamically-typed scripting language to easily explore and develop new algorithms.

Going forward, we designed a new framework that can extend those libraries to perform the computation on heterogeneous platforms by using stream processing concepts.

The purpose is to overcome the computational challenge of the model in order to provide real-time detection, recognition and localization of complex objects in image/video streams.


Example: LAGR a Near-to-far obstacle detection [4]

1. CPU (Central Processing Unit):
   - Easy-to-use for the design of new algorithm
   - Possibility to combine existing machine learning blocks
   - Existence of many libraries to handle data and basic operations
   - Slow for intensive arithmetic operations
   - Low potential for parallelization
   - Best-suited for new algorithms exploration and off-line learning

2. GPU (Graphic Processing Unit):
   - Allow high-throughput computation
   - Libraries and APIs are starting to appear
   - Scalable algorithms for future devices
   - Increasing number of stream processor every 18 months
   - Limitation in the type of function calls
   - Longer developing/debugging time
   - High power consumption
   - Best-suited for High-throughput screening or Fast inference computing

3. FPGA (Field-Programmable Gate):
   - Allow high-throughput computation
   - Low power consumption
   - Ultra-light self-contained platform
   - We designed a Dataflow Architecture with a Dataflow compiler [5]
   - Fixed-point precision (for now)
   - Generality is often traded for efficiency
   - No standard software environment
   - Best-suited for embedded devices where power or weight is a constraint

References