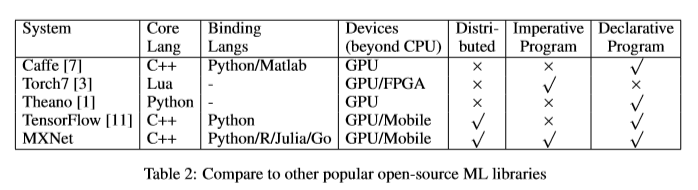
Declarative programming is useful in specifying the computation structure, while imperative programming is more natural for parameter updates and interactive debugging.

Intro:

Different computation and programming models have different emphasis, i.e. imperative programming focuses on calculation details, while declarative programming tends to define the overall model structures. Identifying the merits of mainstream computation and execution models for deep learning system, this paper proposes MXNET, a new machine learning system, with combined advantages, especially for deep learning. MXNET is a lightweight and self-contained system and provides interface for multiple programming language. This system aggressively reduces the consumed memory with memory reuse mechanism and provide distributed synchronization to support data parallelization. The performance evaluation of MXNET is illustrated at the end by comparing with mainstream machine learning systems, such as TensorFlow. Also, the memory consumption is investigated to show the efficiency.

Contribution

1. Support for multiple languages and combination of different programming paradigm
2. Aggressively reduce memory footprint, preforming in-place update and memory space reuse whenever possible
3. A compact communication API, supporting systems running on multiple machines with little change
4. Little weight with no other dependency and with more language supports



Programming Interface

1. Symbol: declarative Symbolic Expressions
2. NDArray for calculation with lazy evaluation to enable backend to resolve data dependency

Lazy evaluation means that any operations on the array (potentially including array construction) are not performed immediately but are delayed until evaluation is specifically requested. Evaluation of only parts of the array is also possible.

1. KVStore: Data Synchronization Over Devices
2. Other Modules: data prefetching, preprocessing with multi-threaded supports

Implementation

Graph Optimization: use only forward graph for output, while for extracting features from internal layers, the last layer can be skipped; well-optimized “big” operations, such as a layer in neural network

Memory Allocation: an ideal allocation strategy requires O(n2) time complexity, 类似LRU? Maybe least internal segment? I guess.