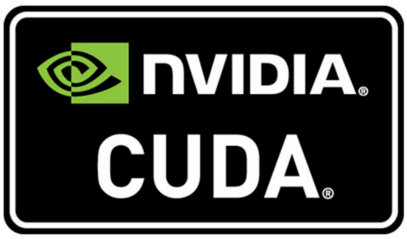
# A Comparison of Some Machine Learning Development Tools

Cameron Ridderikhoff and Matthew Kluk

Last Edited: May 30, 2018

# Contents




# 1 CUDA

## 1.1 Overview:

CUDA is a tool that can be used in many GPU programming circumstances, the main one that we will be discussing in this document is machine learning. The CUDA toolkit has libraries, debugging tools, optimization tools, a compiler and a runtime library, as well as giving access to code samples, programming guides, user manuals, and API references for beginners. CUDA programmers can code in C, C++, Fortran, without any extra software, or they can use Python, and MATLAB through using extensions. Theoretically should perform better than other platforms on Nvidia GPUs, since it is proprietary software made by Nvidia, so CUDA was built for all the features of Nvidia's GPU architecture. CUDA is also made to take full advantage of Nvidia's application parallelism on their cards. CUDA programmers can use CUDA via the driver API, which is a lower level API, or through the runtime API, which is higher level.

## 1.2 Advantages of Using CUDA:

1. Scattered Reads
    - Code can be read from wherever it is stored in memory.
2. Unified Memory
    - All processors share the memory uniformly.
3. Fast Shared Memory
    - The memory can be accessed by multiple processes.
    - FSM can be used as a user managed cache.
4. Fast Communication with the GPU
5. Full Support for Integer and Bitwise Operations

6. Is a Piece of Proprietary Software
7. Free

### 1.3 Limitations of Using CUDA:

1. Performance May Be Reduced When Copying Data Between Host and Device
    - This is due to system bus bandwidth, and latency.
2. Can Only be Used on Nvidia CUDA-Enabled GPUs
3. Threads Should Be Run in Groups of at Least 32
    - The single instruction, multiple data execution model becomes a significant limitation for any inherently divergent task.
    - Branches do not significantly affect performance, as long as the 32 threads branch to the same location.
4. The Kernal Cannot be Compiled at Runtime
5. Rendering Languages Have Access to CUDA Memory, but the Inverse is not True
6. There is no Emulator or Fallback Functionality for CUDA
7. Code is Processed According to C++ Syntax
    - Could lead to code giving errors when using valid syntax.

### 1.4 Used By:

- Nvidia
- Adobe
- Ansys
- Autodesk
- Dassault Systemes
- MathWorks
- Microsoft
- National Instruments
- Wolfram Mathematica


# 2 OpenCL

## 2.1 Overview:

OpenCL is an open-source, royalty free, high level graphics API. It is CUDAs main competition, but struggles to gain the market over its proprietary rival, since many people believe it lacks the speed of CUDA. In most studies, OpenCL performs negligibly slower than, if not as fast as CUDA.One study compared CUDA programs to direct OpenCL translations found CUDA was at most 30% better at performing on Nvidia compilers, but the study said that this discrepancy is mostly attributable to the lack of optimizations that they could have made for the OpenCL code, as well as the optimizations that Nvidia cards have for CUDA vs. OpenCL. However, another study at D-Wave Systems Inc. found that "The OpenCL kernel's performance is between about 13% and 63% slower, and the end-to-end time is between about 16% and 67% slower" than CUDA's performance. OpenCL is mostly used by programmers who want to write portable code that can be used on many different GPUs, rather than exclusively Nvidia graphics cards.

## 2.2 Advantages of Using OpenCL:

1. OpenCL can Use Parallelism With the Following
   - Vector types and operations.
   - Synchronization.
   - Functions that work with items of data and groups of data.
2. Claims to be Easy to Port Due to its Abstracted memory and its Execution Model
3. Allows the Work to be Split Between the CPU and GPU
4. Designed to Work on Most GPUs

## 2.3 Limitations of Using OpenCL:

1. Portability Can Lead to Poor Performance if the Proper Optimizations are not Implemented
2. Code Must be Written in OpenCL C, Which is Based on C99 and C++11
3. Some Useful Functions and Data Types Have Been Omitted
    - Function pointers.
    - Bit fields.
    - Variable-length arrays.
4. Recursion is Forbidden.
5. The Standard C Library is Replaced by a Custom Set of Functions, Which are Built With Mathematical Operations in Mind

## 2.4 Used By:

- Apple Inc.
- IBM
- Samsung
- QUALCOMM
- Altera
- Texas Instruments
- Nvidia
- Marcell
- Vivante
- Creative Labs

# 3 TensorFlow

## 3.1 Overview:

TensorFlow is an open-source machine learning framework, which was built with an emphasis on high performance numerical computation. It is used in neural networking, automated image captioning software, voice recognition, sentimental analysis, threat detection, fraud detection, language detection, time series, and motion detection. The idea that TensorFlow bases itself around is dataflow graphs. These graphs are used to represent computation, shared state, and the operations that mutate that state. The nodes of these graphs can be mapped over many different machines in a cluster, and can use multicore CPUs, GPUs, and special TPUs for mapping out nodes. TensorFlow operates on nodes and the connections between the nodes, called tensors, in order to represent neural networks. In an introductory paper, TensorFlow was, on a single machine, within six percent of Torch's performance, and had shorter step times than Caffe.

## 3.2 Advantages of Using TensorFlow:

1. Supports Python With a Stable API
2. Can Use Nvidia's cuDNN Library to Accelerate Neural Network Computation

## 3.3 Limitations of Using TensorFlow:

1. Java, C, and Go Language Support is Limited

### 3.4 Used By:

- Airbnb
- AMD
- Nvidia
- UBER
- DeepMind
- eBay
- Google
- Snapchat
- Intel
- Twitter
- Coca-Cola
- Lenovo
- IBM Power Systems


## 4 Apache Spark

### 4.1 Overview:

Spark is an open-source, unified, in-memory analytics engine for large scale data processing, but it can also be used for machine learning. Spark uses a directed acyclic graph scheduler, a query optimizer, and a physical execution engine to try and increase performance. Apache Spark was developed in response to the MapReduce model's forced linear

dataflow requirement. Spark's resilient distributed dataset (RDD) oriented model allows the RDDs to function as a working set of programs that offers a deliberately restricted form of distributed shared memory. The base of Spark is Spark Core, which "provides in-memory computing capabilities to deliver speed, a general execution model" and a collection of APIs for Java, Scala, and Python. Core also provides task dispatching, task scheduling, and I/O functionalities by using the RDD model. Spark also comes with MLlib, or Machine Learning Library, which is a distributed machine learning framework on top of Spark Core. Spark claims that MLlib "allows data scientists to focus on their data problems and modles instead of solving the complexities surrounding distributed data". MLlib comes with algorithms for most use cases, but if a specific algorithm is made, it can be included in MLlib, if a community member writes the code. The framework is scalable, from one computer to large clusters of 8000, or potentially more, nodes.

### 4.2 Advantages to Using Apache Spark:

1. Access to SQL and DataFrames, MLlib, GraphX, and Spark Streaming Libraries
2. Can be Run in the Following Ways
    - Standalone mode.
    - On Amazon EC2.
    - On Hadoop YARN.
    - On Apache Mesos.
    - On Kubernetes.
3. Can be Programmed With Scala, Java, Python, and R
4. Uses Implicit Data Parallelism
5. Fault Tolerant
    - This is due to the RDD oriented model, which saves the lineage of each RDD, allowing lost data to be from the RDDs recovered.
6. Supports Iterative Algorithms and Interactive/Exploratory Data Analysis
    - Spark claims a significant latency reduction compared to a MapReduce implementation for these algorithms.

### 4.3 Limitations to Using Apache Spark:

1. Requires a Cluster Manager, and Can Use Hadoop YARN or Apache Mesos
2. Requires a Distributed Storage System, and Can Use Hadoop DFS, Mar-FS, Cassandra, OPenStack Swift, Amazon S3, Kudu, or a Custom Implementation

### 4.4 Used By:

- UC Berkeley AMPLab
- 4Quant
- Act Now
- Adatao Inc.
- AsiaInfo
- eBay
- Groupon
- GraphFlow Inc.
- Istanbul Sehir University
- TripAdvisor
- University of Missouri Data Analytics and Discover Lab
- Yahoo!


## 5 scikit-learn

### 5.1 Overview:

The scikit-learn toolkit is Python based, and is used for machine learning, data mining, and data analysis. It is part of the series of scikit toolkits, which are add-on packages that extend the functionality of scipy, and thus it is built upon, and is compatible with, popular Python libraries like numpy, scipy, and matplotlib. This package is largely written in Python, though some core algorithms are written in Cython for better performance, which means that some Python distributions, like some 64-bit Mac OS Anaconda distributions,

that don't have Cython might have some issues. The focus of scikit-learn is on modelling data, not loading the data in, manipulating it, or summarizing the data. For that, it recommends using other libraries like numpy, pandas, and similar libraries.

## 5.2 Advantages to Using scikit-learn:

1. Features Various Classification, Regression, and Clustering Algorithms
    - Support vector machines.
    - Random forests.
    - Gradient boosting.
    - k-means.
    - DBSCAN.
2. Both Supervised and Unsupervised Learning can be Implemented
3. The Entire Process is Five Steps
    (a) Choose a class of model by importing the appropriate estimator class from Scikit-Learn.
    (b) Choose model hyperparameters by instantiating this class with desired values.
    (c) Arrange data into a features matrix and target vector following the discussion above.
    (d) Fit the model to your data by calling the fit() method of the model instance.
    (e) Apply the Model to new data: For supervised learning, often we predict labels for unknown data using the predict() method. For unsupervised learning, we often transform or infer properties of the data using the transform() or predict() method.

## 5.3 Limitations of Using scikit-learn:

1. Data Mining Capabilities aren't as fully featured as Other non-Python Based Tools
2. Can Only Code in Python

## 5.4 Used By:

- Spotify
- Inria
- betaworks

- Evernote
- Telecom ParisTech
- Change.org
- OkCupid
- INFONEA
- Dataiku
- Otta Group


# 6 MXNet

## 6.1 Overview:

Apache MXNet is an open-source, scalable, portable deep learning framework. This package claims to put a focus on speeding up the development process and the deployment of large-scale deep neural networks. MXNet was programmed in C++, which it uses as a backend, but any of the following can be used for a frontend: Python, R, Scala, Julia, Perl, MatLab, Javascript, C++, or Go.

## 6.2 Advantages of Using MXNet:

1. Contains the Gluon Library, Which Provides a High-Level Interface
    - Apache claims that the Gluon library makes it easy to train dynamic learning maps, without sacrificing training speed.
2. MXNet Offers Multiple Features, Below are Some
    - Device placement.
    - Multi-GPU training.
    - Automatic mathematical differentiation.

- Optimized predefined layers.
- The user can write layers themselves if they prefer.
- Imperative programming support.
- Symbolic programming support.

3. Claims to Have a Near Linear Speedup Across Multiple GPUs
    - Ie. Increasing the number of GPUs by 100% creates a near 100% speedup.

### 6.3 Limitations of Using MXNet:

1. Documentation Leaves Much to be Desired
    - According to Martin Heller, MXNet doesnt say that some example projects need certain prerequisites to make MXNet work with different compilers/non-CUDA drivers, or to work with multiple processor cores in parallel.
    - Heller also notes that most of the examples are Python centered, with few examples in other languages, although updates have brought in more examples in other languages.
2. Lack of Visual Debugging
3. Similar Performance to Any CUDA Based Framework

### 6.4 Used By:

- Amazon - MXNet is their deep kearning framework of choice because "it scales and runs better than almost anything else out there", it has compact size, and cross-platform portability
- Intel
- Microsoft
- Wolfram Research
- MIT
- University of Washington
- Hong Kong University of Science and Technology


# 7 H2O.ai

## 7.1 Overview:

H20 is an open-source, in-memory, distributed and scalable machine learning and predictive analytics platform. H2O was made for big data analysis, such as exploring and analyzing datasets in cloud computing systems, as well as in traditional operating systems. It uses the divide and conquer method for deep learning, wherein the program divides data into subsets and solve each subset in parallel. In the H2O environment, the authors have also created H2O4GPU, which is a piece of software that allows H2O to be used as a drop-in replacement for scikit-learn on selected algorithms. H2O can be called from R or Python, and can be programmed in Java 6 or later, Python 2.7 or later, or 3.5 or later, any R version that is not 3.1.0, or Scala versions 1.4-1.6. It can be run on Windows 7 or later, OS X 10.9 or later, Ubuntu 12.04 or later, RHEL/CentOS 6 or later, Apache Hadoop Distributed File System. However, the H2O GUI is accesed through internet browsers, and is compatible with Chrome, Safari, Firefox, and Internet Explorer.

## 7.2 Advantages of Using H2O.ai:

1. Provides Data Structures Suitable for Working With and Analyzing Large Data Sets
2. Ability to Interrupt the Program and Get an Approximate Solution
    - As opposed to having to wait for the slower, optimized solution.
3. Can be Used on One or More Nodes
4. Implements Fine Grain Parallelism, Which it Claims Increases Performance on a Large Number of Nodes

## 7.3 Limitations of Using H2O.ai:

1. A group of R users felt they couldn't call other machine learning algorithms from R
2. Made by a Private Company, Some Say Its More of a Product Than a Project

3. Documentation isn't Very Good
4. Doesn't Allow the User to Alter Algorithms as Needed

### 7.4 Used By:

- ADP (Automatic Data Processing)
- Change Healthcare
- PayPal
- Capital One
- Comcast
- eBay
- Cisco


## 8 Keras

### 8.1 Overview:

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. It is based around models, which is defined as "a sequence or a graph of standalone, fully-configurable modules that can be plugged together with as little restrictions as possible. In particular, neural layers, cost functions, optimizers, initialization schemes, activation functions, regularization schemes are all standalone modules that you can combine to create new models." All models are defined by Python code, and so Keras is meant to be used with Python along with a backend engine like Tensorflow, Theano, or CNTK. Use of a Tensorflow backend or a Theano backend depends on what hardware you are using, Theano might be better if you only have one GPU, or aren't using GPU acceleration at all, but Tensorflow might work better if you are using multiple threaded GPUs. However, Tensorflow is probably the most frequently updated and used of the machine learning backends.

### 8.2 Advantages of Using Keras:

1. Has a Variety of Different Built-In Layers
2. The User Can Define Layers
3. Can Be GPU-Accelerated by the Nvidia CUDA Deep Neural Network Library
4. Keras Claims That Writing Code is Simple and Straightforward
5. Object Oriented Framework, Where Each Object Has Properties to Manipulate

### 8.3 Limitations of Using Keras:

1. Considered Less Flexible and More Prescriptive Than Other APIs
2. Can Give Frustrating Low-Level Errors
3. Less Customizable Than Other APIs
4. May Be Rather Slow, Since it is a Frontend Framework
5. Lack of Pre-Trained Models

### 8.4 Used By:

- Home61
- Suggestic
- Stylight
- Strike
- PicnicHealth
- Cyanapse
- Katomi
- Advance


# 9 Microsoft CNTK:

## 9.1 Overview:

Microsoft's Cognitive Toolkit is an open-source deep learning toolkit designed for ease of use. It claims to be production ready and to scale to multi-GPU/multi-server networks. Microsoft uses the software for programs like Cortana, Bing, Skype, and for Microsoft Research. The CNTK can be included as a library for Python, C++ and C#, but can also be used as a standalone tool. CNTK can be run on 64-bit Linux and 64-bit Windows. It is useful for tasks like reinforcement learning, supervised or unsupervised learning, and it can also be used to train, with Azure GPU and Azure networks.

## 9.2 Advantages to Using Microsoft CNTK:

1. Has Built-In Components to Handle Multi-Dimensional Dense or Sparse Data
    - FNN
    - CNN (Cellular Neural Network)
    - RNN (Recurrent Neural Network)
    - LSTM (Long Short-Term Memory)
    - Batch normalization
    - Sequence-to-Sequence with attention
2. Parallelism With Accuracy on Multiple GPUs
3. Ability to Add New Core-Components on the GPU From Python
4. Automatic Hyperparameter Tuning
5. Has Built-In Readers Optimized for Massive Datasets

### 9.3 Limitations to Using Microsoft CNTK:

1. Doesn't Support Programming in R
2. Mainly Used for Machine Learning in AI
3. Doesn't Support OS X

### 9.4 Used By:

- Microsoft


## 10 Caffe

### 10.1 Overview:

Caffe is an open-source deep learning framework made with an emphasis on expression, speed, and modularity. Caffe can be run from the command line, but it also has Python and MATLAB interfaces. Caffe is made for convolutional neural network algorithms, but is mostly used for image processing. Caffe is run on top of CUDA, so as such, can only be run on CUDA-compatible hardware. Caffe claims that its expressive architecture encourages application and innovation. The models and their optimization can be defined without having to be hard-coded, which allows the user to switch between the CPU and the GPU by changing one flag.

### 10.2 Advantages to Using Caffe:

1. Supports GPU Training, and Has Many Common Functions, "Out of the Box"
2. Allows Quick Application of Deep Neural Networks to Problems
3. Many Common Functions
4. Large Community and Support for Releases

### 10.3 Limitations to Using Caffe:

1. Not Very Expansive, There are Some Use Cases Caffe Cannot Cover*
2. Only One Output Format: HDF5
3. Multi-GPU training is only Partially Supported*

*This is according to Caffe Pros and Cons (12.3.7 of this document), from 2015, information may have changed.

### 10.4 Used By:

- Facebook
- Nvidia
- Intel
- Sony
- Samsung
- Windows
- Pinterest
- Adobe
- Yahoo!

## 11 Less Relevant Development Tools

### 11.1 lime:

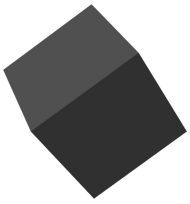
The lime package uses the Local Interpretable Model-Agnostic Explanations technique to explain why machine learning programs give the output that they give. The authors of lime hope this will allow humans to be able to put trust into the black box that is machine learning. They claims that lime is able to explain any black box classifier, with two or more classes. Lime has built in support for scikit-learn classifiers. It uses a local linear approximation of the model's behaviour to create a user readable form of the model's decision function. At the time of writing, lime does not have the ability to explain image classifications.

### 11.2 Metal:

Metal is an low-level proprietary API that is used exclusively for Apple Inc. Metal is mostly used for video games and 3D rendering, and it claims to provide near-direct access to the GPU and to enable the GPU to take more control of the rendering pipeline, such as assigning of resources to shaders, and rendering details. Theoretically, Metal should have better performance than OpenCL, due to its proprietary status. Metal also provides support for GPU-accelerated machine learning.


### 11.3 Microsoft DirectCompute:

DirectCompute is part of Microsoft's DirectX collection of APIs. It runs on GPUs that use either DirectX 10 or DirectX 11, and it shares computational interfaces with OpenCL and CUDA. This API uses a high-level shading language and is designed for photo and video processing, game physics and artificial intelligence, and resource intensive rendering eggects. According to Nvidia's DevZone DirectCompute programming guide, DirectCompute has "efficient interoperability with D3D graphics resources".



### 11.4 Edward:

Edward is a Python library built for probabilistic modeling, inference, and criticism, and it is built on TensorFlow. The library fuses Bayesian statistics and machine learning with deep learning and probabilistic programming. It supports modeling with directed graphical models, neural networks, implicit generative models, Bayesian non-parametrics

and probabilistic programs. Edward also supports inference with variation, and Monte Carlo, as well as compositions of inference. It supports criticism of the model and inference with point-based evaluations and posterior predictive checks.


### 11.5 Intel Nervana's neon:

Nervana's neon is a deep learning framework designed for ease of use and extensibility. It supports convolutional neural networks, recurrent neural networks, long short-term memories, and autoencoders. It also allows for basic automatic mathematical differentiation. The neon framework also has GPU kernel library, and Intel CPU MKLML library integration. It's a framework for visualization, and as such is mostly used for image processing. Can be run on Linux or OS X exclusively, and it runs on Python 2.7, or Python 3.4 or later. According to Stanley Dukor, neon trains networks about twice as fast as other frameworks, and is the fastest framework alive.


### 11.6 Torch:

Torch is an open-source scientific computing framework with support for machine learning algorithms. It claims to be built to maximize the speed and flexibility of scientific algorithms, while making the process of building the algorithms simple. Some of the features that Torch has are: a N-dimensional array, routines of indexing, slicing and transposing, an interface to C, via LuaJIT, linear algebra routines, GPU support, neural network, and energy-based models.


### 11.7 Vulkan:

Vulkan is an extremely low-level 3D graphics and computing API that focusses on real-time 3D graphics targets, such as video games and interactive media. It works on having a more balanced CPU/GPU usage, and can distribute work amongst multiple CPU cores.


### 11.8 Eclipse Deeplearning4j:

Eclipse Deeplearning4j is a software distribution of several projects targeted at integrating with enterprise environments. Deeplearning4j wants to enable developers and large organizations to build deep learning applications covering the whole deep learning workflow from data preprocessing through distributed training and hyperparameter optimization and production-grade deployment. The software focusses on trying to bridge to gap between research and real world applications, by facilitating the process of building an application without having to rely on third-party providers for ETL libraries, or tensor libraries. Deeplearning4j provides features like a neural network DSL, n-dimensional arrays for Java, a tensor library, an ETL library, a C++ library for tensor operations, a Python interface for the tensor library to integrate with Numpy, and automatic tuning of neural networks via a hyperparameter search.

# 12 Links and References

## 12.1 Images:

CUDA OpenCL Metal Microsoft Logo TensorFlow Apache Spark MXNet Microsoft CNTK Edward H2O Caffe Intel Nervana's neon Torch Eclipse Deeplearning4j Vulkan scikit-learn Keras

## 12.2 References:


1. CUDA: Website Wikipedia
2. OpenCL: Website Wikipedia
3. Metal: Website Wikipedia
4. DirectCompute: Wikipedia Programming Guide
5. TensorFlow: Website Wikipedia
6. Apache Spark: Website Wikipedia
7. Apache MXNet: Website Wikipedia
8. Microsoft CNTK: Website Wikipedia
9. lime: Website Promotional Video
10. Edward: Website
11. H2O: Website Wikipedia
12. Caffe: Website Wikipedia
13. Intel Nervana's neon: Website
14. Torch: Website Wikipedia
15. Eclipse Deeplearning4j: Website Wikipedia
16. Vulkan: Website Wikipedia
17. scikit-learn: Website Wikipedia
18. Keras: Website Wikipedia

### 12.3 Articles:

1. Comparison Between CUDA and OpenCL: 1 2 3
2. TensorFlow: A System for Large-Scale Machine Learning
3. Martin Heller's Review of MXNet
4. MXNet is "blazingly fast"
5. Li Yin Sulimowicz' Review of Deep Learning Tools
6. Python Deep Learning Frameworks Reviewed
7. Caffe Pros and Cons
8. Neon is the "fastest framework alive"
9. Scikit-learn: The five step method, The focus of scikit-learn
10. DirectCompute Programming Guide