

Deep Learning Resource Matrix

Framework	Description	Background	Use Cases	Supports
TensorFlow	<ul style="list-style-type: none"> • Open source software library for numerical computation using data flow graphs. • Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. • The system is general enough to be applicable in a wide variety of other domains as well. 	<ul style="list-style-type: none"> • Developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research. • More information <ul style="list-style-type: none"> • https://www.tensorflow.org 	<ul style="list-style-type: none"> • Fit a neural network on MNIST handwritten digit recognition data. • Note: MNIST is a simple computer vision dataset • Breathe life into older media – “Remastering Classic Films in TensorFlow with Pix2Pix” 	<ul style="list-style-type: none"> • General numerical computations on CPU's, GPU's and GPU clusters • Flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.
Theano	<ul style="list-style-type: none"> • Python library that lets you to define, optimize, and evaluate mathematical expressions, especially ones with multi-dimensional arrays (numpy.ndarray). • Using Theano, it's possible to attain speeds rivaling hand-crafted C implementations for problems involving large amounts of data. It can also surpass C on a CPU by many orders of magnitude by taking advantage of recent GPUs. • Theano combines aspects of a computer algebra system (CAS) with aspects of an optimizing compiler. • Also generates customized C code for many mathematical operations. This combination of CAS with optimizing compilation is particularly useful for tasks in which complicated mathematical expressions are evaluated repeatedly and evaluation speed is critical. For situations where many different expressions are each evaluated once Theano can minimize the amount of compilation/analysis overhead, and still provide symbolic features such as automatic differentiation. 	<ul style="list-style-type: none"> • Theano was written at the LISA lab to support rapid development of efficient machine learning algorithms. Theano is named after the Greek mathematician, who may have been Pythagoras' wife. Theano is released under a BSD license (link). • More information <ul style="list-style-type: none"> • http://deeplearning.net/software/theano/ 	<ul style="list-style-type: none"> • Fit a logistic regression model on a synthetic dataset with GPU acceleration. 	<ul style="list-style-type: none"> • Tight integration with NumPy – Use <code>numpy.ndarray</code> in Theano-compiled functions. • Transparent use of a GPU – Perform data-intensive calculations up to 140x faster than with CPU.(float32 only) • Efficient symbolic differentiation – Theano does your derivatives for function with one or many inputs. • Speed and stability optimizations – Get the right answer for $\log(1 + x)$ even when x is really tiny. • Dynamic C code generation – Evaluate expressions faster. • Extensive unit-testing and self-verification – Detect and diagnose many types of errors.

<p>Caffe</p>	<ul style="list-style-type: none"> • Deep learning framework designed with expression, speed, and modularity in mind. 	<ul style="list-style-type: none"> • Developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors. Yangqing Jia created the project during his PhD at UC Berkeley. Caffe is released under the BSD 2-Clause license • More information <ul style="list-style-type: none"> • http://caffe.berkeleyvision.org/ 	<ul style="list-style-type: none"> • Academic research projects • Startup prototypes • Large-scale industrial applications in: <ul style="list-style-type: none"> • Vision • Speech • Multimedia • Fit a neural network on MNIST handwritten digit recognition data. 	<ul style="list-style-type: none"> • Expressive architecture encourages application and innovation. Models and optimization are defined by configuration without hard-coding. Switch between CPU and GPU by setting a single flag to train on a GPU machine then deploy to commodity clusters or mobile devices. • Extensible code fosters active development. In Caffe's first year, it has been forked by over 1,000 developers and had many significant changes contributed back. Thanks to these contributors the framework tracks the state-of-the-art in both code and models. • Speed makes Caffe perfect for research experiments and industry deployment. Caffe can process over 60M images per day with a single NVIDIA K40 GPU*. That's 1 ms/image for inference and 4 ms/image for learning. We believe that Caffe is the fastest convnet implementation available. • Community: Caffe already powers academic research projects, startup prototypes, and even large-scale industrial applications in vision, speech, and multimedia. Join our community of brewers on the caffe-users group and Github. With the ILSVRC2012-winning SuperVision model and caching IO. Consult performance details.

MXNet

- Open-source deep learning framework that allows you to define, train, and deploy deep neural networks on a wide array of devices, from cloud infrastructure to mobile devices.
- It is highly scalable, allowing for fast model training, and supports a flexible programming model and multiple languages. MXNet allows you to mix symbolic and imperative programming flavors to maximize both efficiency and productivity.
- MXNet is built on a dynamic dependency scheduler that automatically parallelizes both symbolic and imperative operations on the fly.
- A graph optimization layer on top of that makes symbolic execution fast and memory efficient.
- The MXNet library is portable and lightweight, and it scales to multiple GPUs and multiple machines.

- Developed by collaborators from multiple universities and companies.

• More information

- <http://mxnet.io/>

• Basics

- Manipulating arrays, building networks, loading and preprocessing data

• IO

- Data loading, Image IO, Record IO, Custom Image IO

• Computer Vision

- Use machine learning to modify, classify, and segment images and video
- Handwritten Digit Classification
- Image Classification
- Image Segmentation
- Object Detection using faster R-CNN
- Neural Art: Adding Artistic Style to Images
- Large Scale Image Classification
- Classifying Real-World Images with a Pre-Trained Model

• Natural Language Processing

- Applications that understand, generate, and summarize text-based data
- Character Level LSTM
- Text Classification using Convolutional Neural Network
- NCE Loss

• Speech Recognition

- Applications that map natural speech to text
- Speech LSTM
- Baidu Warp CTC

• Generative Networks

- General Adversarial Network

• Unsupervised Machine Learning

- Develop applications for discovering existing structures and relationships in datasets
- Matrix Factorization
- Auto Encoders

- **Flexible** – Supports both imperative and symbolic programming
- **Portable** – Runs on CPUs or GPUs, on clusters, servers, desktops, or mobile phones
- **Multiple Languages** – Supports multiple languages, including C++, Python, R, Scala, Julia, Matlab and Javascript - All with the same amazing performance.
- **Auto-Differentiation** – Calculates the gradient automatically for training a model
- **Distributed on Cloud** – Supports distributed training on multiple CPU/GPU machines, including AWS, GCE, Azure, and Yarn clusters

			<ul style="list-style-type: none"> • Recommendation Systems • Visualization <ul style="list-style-type: none"> • Understanding the vanishing gradient problem through visualization • Embedded <ul style="list-style-type: none"> • Develop machine learning applications running on embedded devices, such as the Raspberry Pi 	
Apache SystemML (incubator project)	<ul style="list-style-type: none"> • Flexible machine learning system that automatically scales to Spark and Hadoop clusters. 	<ul style="list-style-type: none"> • Created in 2010 by researchers at the IBM Almaden Research Center led by IBM Fellow Shivakumar Vaithyanathan. • IBM was open-sourcing SystemML as part of IBM's major commitment to Apache Spark and Spark-related projects. SystemML became publicly available on GitHub on August 27, 2015 and became an Apache incubator project on November 2, 2015. • More information <ul style="list-style-type: none"> • https://systemml.apache.org/ 	<ul style="list-style-type: none"> • Linear algebra, matrices, and statistical functions 	<ul style="list-style-type: none"> • Algorithm customizability via R-like and Python-like languages. • Multiple execution modes, including Standalone, Spark Batch, Spark MLContext, <u>Hadoop</u> Batch, and JMLC. • Automatic optimization based on data and cluster characteristics to ensure both efficiency and scalability.
BigDL	<ul style="list-style-type: none"> • Distributed deep learning library for Apache Spark. • Users can write their deep learning applications as standard Spark programs, which can directly run on top of existing Spark or Hadoop clusters. 	<ul style="list-style-type: none"> • More information <ul style="list-style-type: none"> • https://github.com/intel-analytics/BigDL 	<ul style="list-style-type: none"> • Analyze a large amount of data on the same Big Data (Hadoop/Spark) cluster where the data are stored (in, say, HDFS, HBase, Hive, etc.). • Add deep learning functionalities (either training or prediction) to your Big Data (Spark) programs and/or workflow. • Leverage existing Hadoop/Spark clusters to run your deep learning applications, which can be then dynamically shared with other workloads (e.g., ETL, data 	<ul style="list-style-type: none"> • Rich deep learning support. Modeled after Torch, BigDL provides comprehensive support for deep learning, including numeric computing (via Tensor) and high level neural networks; in addition, users can load pre-trained Caffe or Torch models into Spark programs using BigDL. • Extremely high performance. To achieve high performance, BigDL uses Intel MKL and multi-threaded programming in each Spark task. Consequently, it is orders of magnitude faster than out-of-box open source Caffe, Torch or TensorFlow on a single-node Xeon (i.e., comparable with mainstream GPU).

			warehouse, feature engineering, classical machine learning, graph analytics, etc.)	<ul style="list-style-type: none"> • Efficiently scale-out. BigDL can efficiently scale out to perform data analytics at "Big Data scale", by leveraging Apache Spark (a lightning fast distributed data processing framework), as well as efficient implementations of synchronous SGD and all-reduce communications on Spark.
DistBelief	<ul style="list-style-type: none"> • Framework for training deep neural networks. 	<ul style="list-style-type: none"> • <i>DistBelief</i> was first presented in the 2012 paper "<i>Large Scale Distributed Deep Networks</i>" by <i>Dean et al.</i> In this paper, the GraphLab distributed computing framework was used to implement <i>DistBelief</i>. • More information <ul style="list-style-type: none"> • https://github.com/alexminnaar/AkkaDistBelief • http://alexminnaar.com/implementing-the-distbelief-deep-neural-network-training-framework-with-akka.html 	<ul style="list-style-type: none"> • Supervised Learning <ul style="list-style-type: none"> • Time Series & Clustering • Bayesian Inference NLP • Distributed Binary Search Tree with Akka 	<ul style="list-style-type: none"> • Avoids GPUs entirely and performs parallel computing with clusters of commodity machines. • Relies heavily on asynchronous message passing which makes the Akka actor framework a suitable alternative