A Survey of
Deep-learning Frameworks

Aniruddha Parvat

Dept. of Computer Engineering,
Sinhgad Institute of Technology,
Lonavala 410401, Maharashtra, India
aniruddhaparvat@gmail.com

Jai Chavan

Dept. of Computer Engineering,
Sinhgad Institute of Technology,
Lonavala 410401, Maharashtra, India
jaipmohite@gmail.com

Siddhesh Kadam

Dept. of Computer Engineering,
Sinhgad Institute of Technology,
Lonavala 410401, Maharashtra, India
sid.kadam19@gmail.com

Souradeep Dev

Dept. of Computer Engineering,
Sinhgad Institute of Technology,
Lonavala 410401, Maharashtra, India
souradeep15@gmail.com

Vidhi Pathak

Dept. of Computer Engineering,
Sinhgad Institute of Technology,
Lonavala 410401, Maharashtra, India
vidhipathak74@gmail.com

Abstract—**Deep learning is a model of machine learning loosely based on our brain. Artificial neural network has been around since the 1950s, but recent advances in hardware like graphical processing units (GPU), software like cuDNN, TensorFlow, Torch, Caffe, Theano, Deeplearning4j, etc. and new training methods have made training artificial neural networks fast and easy. In this paper, we are comparing some of the deep learning frameworks on the basis of parameters like modeling capability, interfaces available, platforms supported, parallelizing techniques supported, availability of pre-trained models, community support and documentation quality.**

Keywords–Machine learning; Deep learning; Neural networks; Software libraries

I. INTRODUCTION

Machine learning is a branch of artificial intelligence that gives computers the ability to learn without being explicitly programmed. Machine learning powers many aspects of our lives from web searches to product recommendation to content filtering and is present in consumer products like smartphones. Machine learning systems are used to identify the content of images and videos, translate speech to text, refine search results, recommend products or posts according to users interests. Deep learning is a branch of machine learning that has some of the best results in these fields. Constructing a pattern-recognition or machine learning system using conventional machine-learning techniques required considerable domain expertise to design a feature extractor to transfer the raw training data into suitable internal representation from which the system could detect or classify patterns in the input data. Representation learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification. Deep learning methods are representation learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. With the composition of enough such transformations, very complex functions can be learned [1].

Deep learning techniques are currently state-of-the-art at pretty much every AI-related application and incremental advances in a field that used to take years to achieve is now happening faster. One instance of that is the advances made in a field by deep-learning is the example of ImageNet [2].

ImageNet is a large dataset of images containing objects whose positions and labels have been labeled by humans. ImageNet contains 16 million images and 20,000 classes. Every year computer vision algorithms are compared against each other by their accuracy. Error rate has dropped drastically since the introduction of deep-learning algorithms into the competition. In 2012, Geoff Hinton and his team used a Convolutional Neural Network that learned its own features and got error rate down to 18.9% [3]. In 2014, Google's Deep Learning algorithm, called GoogLeNet named after Yann Le Cunn, one of the pioneers of Deep Learning research got the error down to 6.7%.Deep learning has revolutionized Speech recognition. All major speech recognition systems such as Skype translator, Google Now, Apple Siri, Microsoft Cortana are all based on deep learning models. Natural language models have been implemented using deep learning since the early 2000s. Recurrent neural networks are the most appropriate for sequential data such as language. Deep learning architectures have achieved state-of-the-art results in many natural language processing tasks such as sentiment analysis [4], spoken language understanding [5], information retrieval, machine translation, contextual entity linking, and others. E-commerce websites have used deep-learning for product recommendations. Deep learning models are also used in fields like face recognition drug discovery and toxicology, biomedical informatics, weather forecasting, finance, etc [6].

Comparison parameters used in this article are:

- Platform: Cross-platform frameworks are preferable, platform include operating system supported, cloud service support.

- Interface: Longer the list of interfaces, more accessible the framework. Interface in mainstream language are preferred.

- Model ling capability: It is the capability of framework to train different type of networks. We have considered three types:

  – Recurrent nets: Recurrent neural networks have been made popular by Juergen Schmidhuber and Sepp Hochreiter [7]. RNNs have a feedback loop where the nets output is fed back into the net along with the next input. Applications of RNNs range from time series analysis to self-driving cars.

  – Convolutional nets: CNNs were pioneered by Yann LeCun at New York University. They are very useful in the field of image, object and speech recognition.

  – Boltzmann Machine: RBMs were created by Geoff Hinton of the University of Toronto. RBMs have applications in dimensionality reduction, classification, feature learning etc.

They can be trained in either supervised or unsupervised way depending on the task [8].

  – Deep Belief nets: DBF were also created by Geoff Hinton. DBNs can be trained one layer at a time [9].

- Support for CUDA, OpenMP, OpenCL.

- Support for pre-trained models: Training a deep neural network takes a lot of time. Some people release their final checkpoint which other people can use for fine-tuning. Pre-trained models are also used for transfer learning. Modify previously trained networks can be used to reduce training epochs [10].

II. LIBRARIES

A. NVIDIA cuDNN

The NVIDIA CUDA Deep Neural Network library (cuDNN) is a GPU-accelerated library of primitives for deep neural networks. The cuDNN provides highly tuned implementations for standard routines such as forward and backward convolution, pooling, normalization, and activation layers. Many frameworks like Caffe, TensorFlow, Theano, Torch rely on high performance GPU acceleration. Fig 1 shows the interaction between hardware and deep learning applications.

B. Theano

Theano was created by the machine learning group at University of Montreal. The head of the group Yoshau Bengio is one of the pioneers of deep learning. It is a cross-platform open source python library [11]. In Theano neural networks and data are represented as matrices and all operations are defined as matrix calculations. Vectorized code can run quickly since multiple values can are processed in parallel. Since Deep Nets require large amounts of computation throughout the training process, vectorization is a highly recommended feature While using theano the user must code every aspect of a net including the nodes, the layers, the activation and the training rate. Therefore all types of deep nets can be trained.Due to its design as a vector processing library, theano is a general use machine learning library and not just a deep learning library. This also makes theano highly extensible. Many libraries extend theano like Lasagne, Keras, Blocks for parameterizing theano functions, Passage for text analysis. These libraries makes using theano for building deep nets easy.

Theano has CUDA and OpenMP support. Keras provides pre-trained VGG networks, and there are scripts to convert Caffe models.
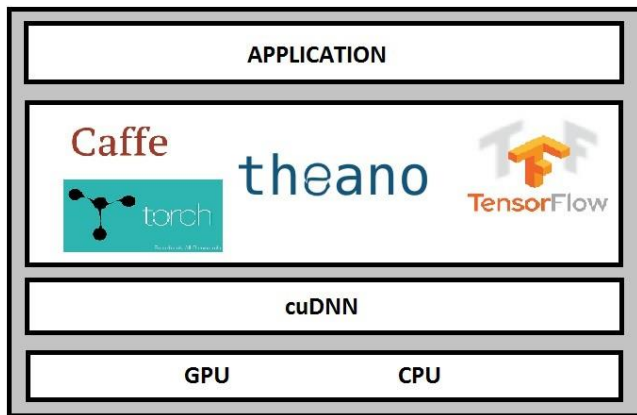
Figure. 1   LIBRARY ARCHITECTURE

Theano has simple but powerful documentation with lot of examples. Keras and Lasagne are really community driven and due to Theano's simplicity to develop new things they are really fast at implementing new features and concepts as they are discovered. The documentation of Keras and Lasagne are very extensive, and each important concept is explained with an example.

C. Deeplearning4j

Deeplearning4j is an open source deep learning framework written in Java, Scala, CUDA, C, C++. It is released under Apache license 2.0. It is developed by a machine learning group led by Adam Gibson and supported by a startup company called Skymind. This deep learning framework works on operating systems like Linux, OS X, Windows, Android.

Deeplearning4j is a commercial grade library that can run on distributed multinode setup. This library is written for java and java virtual machine (JVM). This library also provides GPU support for distributed framework. As a result, the library can run on both Scala and Clojure. Deeplearning4j supports all of the deep nets like restricted Boltzmann machine, deep belief nets, deep autoencoder, recursive neural tensor network, etc. It also includes a vectorization library called Canova which was built by the same team.

It uses iterative map reduce method to train a model on a distributed platform. It is portable and platform neutral, rather than being tied to any cloud service like Azure, AWS or Google Cloud.

In Deeplearning4j parallelism is automatic. It automates the setup of worker nodes, which allows users to bypass libs while creating a parallel network on Spark, Hadoop or

with Akka and AWS. Deeplearning4j is best suited for solving problems and doing it quickly. It supports OpenMP and   its OpenCL support is still being developed. It has been optimized to run on various chips such as x86 and GPU   with CUDA C.

Deeplearning4j supports pre-trained models. The repository of these models is called model zoo. Model Zoo consists   of AlexNet, LeNet, VGGNetA, VGGNetD, GoogLeNet. There are still many pre-trained models which are being added to their model Zoo. The deeplearning4j website provides good documentation and examples.

D. Caffe

Caffe is an open source deep learning toolkit that was developed by Berkeley vision and learning center, and by community contributors. It has an expressive architecture with expression, speed and modularity. Yangqing Jia of UC Berkeley created this project during his Ph.D. It is a cross-platform which can be written in C++, Python and supports Ubuntu, OS X, unofficial Android port and other platforms [12].

Many extensions are actively being added and is still the most popular toolkit used for image classification. The advantage of Caffe is that instead of writing the code, models and optimizations are defined as plaintext schemas. OpenMP is not supported by Caffe, but it supports CUDA and third party implementation of OpenCL.

Caffe supports recurrent and Convolution Nets. Pycaffe interface is used which is an alternative to the command line interface. ResNet-50, ResNet-101, ResNet-152 are the pre-trained models on ImageNet. BLVC AlexNet and BLVC GoogLeNet are some other pre-trained models.

E. Torch

Torch is an open source deep learning framework and is based upon Lua scripting language which is fast, portable and easy to use. It provides a Matlab-like environment for deep learning which provides a wide range of machine learning algorithms. Originally it was developed at NYU by Ronan Collabert, along with Koray Kavukcuoglu and Clement Farabet. It is available on operating systems like Android, Linux, Windows, Mac OS X, iOS and is written in C, C++, Lua.

It includes packages for neural networks, graphical models, image processing, optimization and energy based models. Torch is used for large scale machine learning applications like speech, video and image applications. It has plenty of great extensions and a support of a large community. It offers fast and efficient GPU support, the

option to configure Deep-net parameters and other useful features. The Torch library provides a powerful vectorized implementation of the mathematics behind deep learning algorithms. It has a powerful and flexible N-dimensional array or Tensors which supports transposing, cloning, indexing, slicing, etc. Arbitrary graphs of neural networks can be built and parallelize over CPUs and GPUs.

Torch has various types of extended libraries like CuTorch, NN, Cephes, Dp, NNGraph provide special features which are useful for building a deep net. The CuTorch provides GPU support, the NNGraph help in building different networks and can stack different nets together, the Cepehes extends Torch with a specialized math library, DP is used for streamlining research and development process, NNGraph provides graph tools for the NN package. Many networks are trainable using Torch such as Recurrent, Convolution, Deep belief networks, etc.

It supports OpenMP, OpenCL (Third party implementation) and CUDA [13]. Third party pre-trained models are available on data sets like ImageNet and MNIST. The drawback of Torch is that documentation is patchy and inconsistent.

F. TensorFlow

TensorFlow is an open source library for numerical computation created by Google brain team [14]. TensorFlow grew of another deep net library called DistBelief. It is based on computational graph. A computational graph is a directed graph where nodes represent mathematical operations and edges represent the flow of data between nodes. The name of the library is based on how tensors across the network. Tensors are a type of multidimensional array. This structure of library makes it more than just a deep learning library. Any domain where computation can be modeled as a data flow graph can use TensorFlow.

TensorFlow is written with a Python API over a C/C++ engine, this makes it run faster. TensorFlow has CUDA support. Almost all kind of networks can be built using TF, although it does not allow for hyper-parameter configuration of deep nets. TensorFlow also provides an interface for C++.

TensorFlow is available on Linux and Mac OS X but support for Windows is on the roadmap.   It is also available on mobile and embedded platforms like Android iOS and Raspberry Pi. It has the capability to build large scale machine learning models that can be deployed on variety of platforms ,from smartphones to large clusters consisting a number of nodes and GPUs.

An important new feature is the implementation of data parallelism, which is similar to the Iterative Map-Reduce from Deeplearning4j. It also implements model parallelism, where different portions of the graph can be trained on multiple devices in parallel.

The TensorFlow team has also released TF-Slim which is a high level library to define complex models in TensorFlow. The TF-Slim library provides common abstractions which enable users to define models quickly and concisely, while keeping the model architecture transparent and its hyper-parameters explicit.

The Inception-V3, a state-of-the-art image classification model was built on TF-slim. TensorFlow has visualization tool called TensorBoard for understanding, debugging and optimization of TensorFlow programs. TensorBoard operates by reading TensorFlow events files, which contain summary data that you can generate when running TensorFlow.

TensorFlow is a fast moving project. There are many new features on the roadmap like OpenCL and Windows support, improvement on non-Python language support like C and C++ and some speed and performance improvement.

The documentation for TensorFlow is very extensive and useful; it includes everything from basic installation to building large networks on distributed environment. The TF community is one of the most active as TF is one of most famous (forked and stared) repository on GitHub. It also hosts pre-trained models like autoencoders, Inception (CNN for computer vision), some image classification models in TF-Slim , im2txt and many more on GitHub.

G. Other frameworks

This section is for non mainstream libraries. These libraries sometimes extend other libraries.

Keras: Keras was initially developed as part of the research effort of project ONEIROS. Keras is a high level non mainstream neural network library. It is written in Python and can run on top of either Theano or TensorFlow. It allows fast and easy prototyping. It supports CNNs and recurrent networks and runs seamlessly on GPU.
Due to the design of Theano and TensorFlow it is possible to write high level libraries like Keras which can run on either backend. Keras programs are generally smaller than the equivalent TensorFlow and Theano programs. Fig. 2 shows the architecture of Keras.
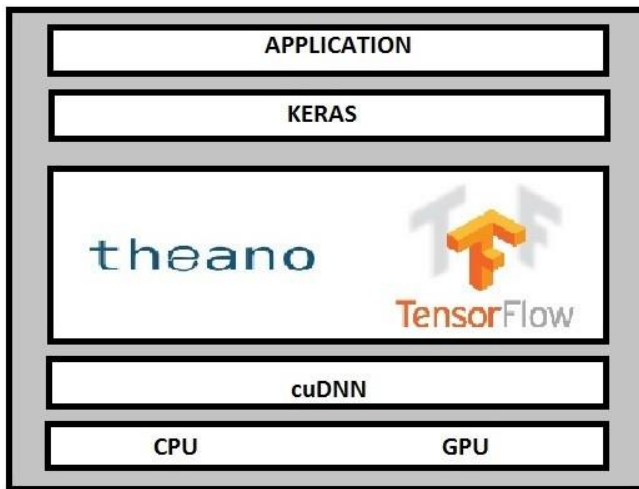
Figure. 2 HIGH LEVEL LIBRARY ARCHITECTURE

1) Lasagne: Lasagne is a lightweight library to build and train neural networks in Theano. It supports CNNS, recurrent networks including Long Short-Term Memory (LMST). Its design is based on six principles: simplicity, transparency, modularity, pragmatism, restraint, focus.

2) DSSTNE: Amazon released DSSTNE (Deep Scalable Sparse Tensor Network Engine) on Github under an open- source Apache license. Amazon released DSSTNE so that deep learning can be extended beyond the image, speech, NLP system. It is written in C++. It can scale out to use multiple GPUs. Amazon has used it to generate personalized product recommendation system.

III. CONCLUSION

We studied some of the top deep learning libraries namely Theano, Deeplearning4j, Caffe, Torch and TensorFlow. Concluding one library is better than others is hard and impractical as these are very active open source projects which get frequent updates. Here are our observations:

- TensorFlow is a very flexible library as it are more than just a deep learning frameworks. Implementing new and non-standard architectures is possible. It is also flexible in terms of employing the variety of devices parts of a computational graph. TensorBoard is a very powerful debugging and visualization tool unique to TensorFlow.
- Theano has libraries like Keras, Lasagne, Block which are very helpful for creating models quickly. The symbolic differentiation feature of Theano is very useful while implementing non-

standard deep architectures.
- TensorFlow and Theano are the most flexible libraries.
- Torch has lots of pre-trained models available. Torch runs on LuaJIT, which is very fast but, Lua is not a mainstream language. Torch documentation needs an update.
- Caffe is useful for image processing, and it has many pre-trained convolutional models hosted on its Model Zoo site.
- Deeplearning4j is the only mainstream framework with interfaces in Java and Scala. It also supports training models on Spark cluster. It provides the best support for distributed training.
- TensorFlow also has the ability to train on a distributed environment; it can employ homogeneous/ heterogeneous devices for training various parts of the graph.

Table. 1 , shows the comparison of the deep learning libraries on the basis of parameters like modeling capability, interfaces available, platforms supported, parallelizing techniques supported, availability of pre-trained models, community support and documentation quality.

TABLE 1. COMPARISON OF LIBRARIES

	Creator	Platform	Interface	Written in	Parallelizing technique support	Modeling capability	Has pre-trained models	Parallel execution
Theano	Universit de Montral	Cross-Platform	Python	Python	OpenMP, CUDA	RNNs, CNNs, RBM, DBNs	Lasagne Zoo	multiple GPUs
Deeplearning4j	Skymind engineering team	Cross-platform	Java, Scala, Clojure	C, C++	OpenMP, CUDA	RNNs, CNNs, RBMs, DBNs	Yes	Yes
Caffe	Berkeley Vision and Learning Center	Ubuntu, OS X, AWS, unofficial port for windows and android	C++, Com-mandline, Python, MATLAB	C++	CUDA, OpenMP (third party implementation)	RNNs, CNNs, RBMs, DBNs	Yes	Yes
Torch	Ronan Collobert, Koray Kavukcuoglu, Clement Farabet	Linux, Android, iOS, Win-dows, Mac OS	Lua, LuaJIT, C, C++	C, Lua	CUDA, OpenMP, OpenCL (third party implementation)	RNNs, CNNs, RBMs, DBNs	Yes	Yes
TensorFlow	Google Brain team	Linux, Mac OS X,	Python, (C/C++ public API only for executing graphs	C++, Python	CUDA	RNNs, CNNs, RBMs, DBNs	Yes	Yes

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol.521, no. 7553, pp. 436–444, 2015.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 248–255.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

[4] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic composition- ality over a sentiment treebank," in Proceedings of the conference on empirical methods in natural language processing (EMNLP), vol. 1631. Citeseer, 2013, p. 1642.

[5] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani- Tur, X. He, L. Heck, G. Tur, D. Yu, and G. Zweig, "Using recurrent neural networks for slot filling in spoken language understanding," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 3, pp. 530–539, March 2015.

[6] Q. V. Le, "Building high-level features using large scale unsupervised learning," in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, May 2013, pp. 8595–8598.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov 1997.

[8] G. Hinton, "A practical guide to training restricted boltzmann ma- chines," Momentum, vol. 9, no. 1, p. 926, 2010.

[9] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural computation, vol. 18, no. 7, pp. 1527– 1554, 2006.

[10] T. Chen, I. Goodfellow, and J. Shlens, "Net2net: Accelerating learning via knowledge transfer," arXiv preprint arXiv:1511.05641, 2015.

[11] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Des- jardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: A cpu and gpu math compiler in python," in Proc. 9th Python in Science Conf, 2010, pp. 1–7.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in Proceedings of the 22nd ACM interna- tional conference on Multimedia. ACM, 2014, pp. 675–678.

[13] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in BigLearn, NIPS Workshop, no. EPFL-CONF-192376, 2011.

[14] T. T. D. Team, R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov et al., "Theano: A python framework for fast computation of mathematical expressions," arXiv preprint arXiv:1605.02688, 2016.