5 Amazing Deep Learning Frameworks Every Data Scientist Must Know! (with Illustrated Infographic)

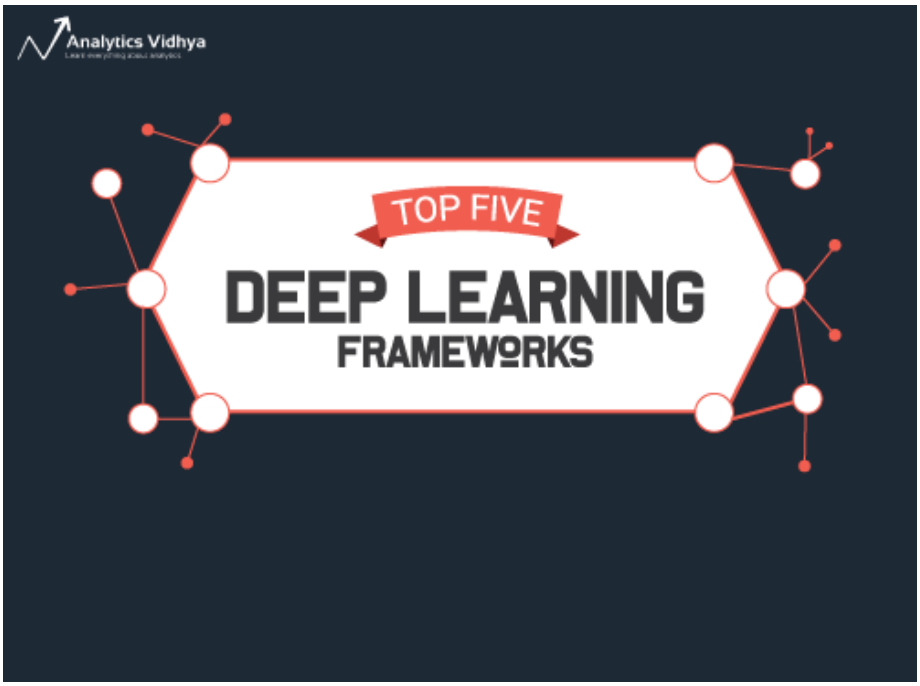
PULKIT SHARMA,

Introduction

I have been a programmer since before I can remember. I enjoy writing codes from scratch – this helps me understand that topic (or technique) clearly. This approach is especially helpful when we're learning data science initially.

Try to implement a neural network from scratch and you'll understand a lot of interest things. But do you think this is a good idea when building deep learning models on a real-world dataset? It's definitely possible if you have days or weeks to spare waiting for the model to build.

For those of us who don't have access to infinite computational resources, you've come to the right place.



Here's the good news – we now have easy-to-use, open source deep learning frameworks that aim to simplify the implementation of complex and large-scale deep learning models. Using these amazing frameworks, we can implement complex models like convolutional neural networks in no time.

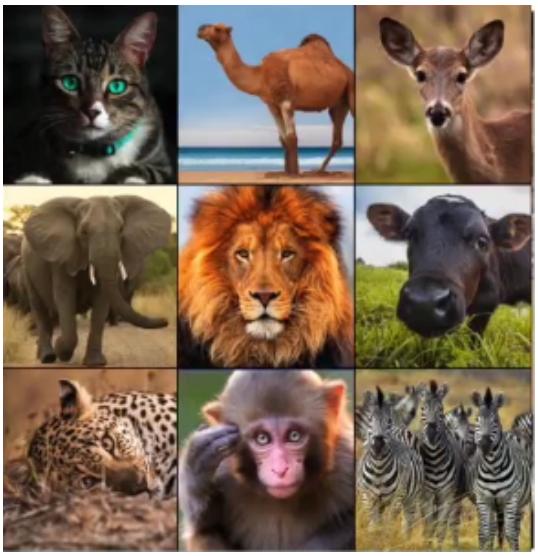
In this article, we will look at 5 super useful deep learning frameworks, their advantages, and their applications. We'll compare each framework to understand when and where we can use each one.

We have created a really cool infographic as well which espouses the value of each deep learning framework. It's available at the end of this article and is a must-have for every data scientist.

Table of Contents

What is a Deep Learning Framework?

Let's understand this concept using an example. Consider the below collection of images:



There are various categories in this image – Cat, Camel, Deer, Elephant, etc. Our task is to classify these images into their corresponding classes (or categories). A quick Google search tells us that Convolutional Neural Networks (CNNs) are very effective for such image classification tasks.

So all we need to do is implement the model, right? Well, if you start to code a Convolutional Neural Network from scratch, it will take days (or even weeks) until you get a working model. We can't afford to wait that long!

This is where deep learning frameworks have truly changed the landscape.

> *A deep learning framework is an interface, library or a tool which allows us to build deep learning models more easily and quickly, without getting into the details of underlying algorithms. They provide a clear and concise way for defining models using a collection of pre-built and optimized components.*

Instead of writing hundreds of lines of code, we can use a suitable framework to help us to build such a model quickly. Below are some of the key features of a good deep learning framework:

1. Optimized for performance
2. Easy to understand and code
3. Good community support
4. Parallelize the processes to reduce computations
5. Automatically compute gradients

These is the criteria I used to pick out my top 5 deep learning frameworks. Let's dive into each of them in detail.

TensorFlow

TensorFlow was developed by researchers and engineers from the Google Brain team. It is far and away the most commonly used software library in the field of deep learning (though others are catching up quickly).

> *The two things I really like about TensorFlow – it's completely open source and has excellent community support. TensorFlow has pre-written codes for most of the complex deep learning models you'll come across, such as Recurrent Neural Networks and Convolutional Neural Networks.*

One of the biggest reasons TensorFlow is so popular is it's support for multiple languages to create deep learning models, such as Python, C++ and R. It has proper documentations and walkthroughs for guidance (did you really expect shabby work from Google?).

There are numerous components that go into making TensorFlow. The two standout ones are:

1. **TensorBoard:** Helps in effective data visualization using data flow graphs
2. **TensorFlow:** Useful for rapid deployment of new algorithms/experiments

The flexible architecture of TensorFlow enables us to deploy our deep learning models on one or more CPUs (as well as GPUs). Below are a few popular use cases of TensorFlow:

1. Text-based applications: Language detection, text summarization
2. Image recognition: Image captioning, face recognition, object detection
3. Sound recognition
4. Time series analysis
5. Video analysis

There are many more use cases. If you have used TensorFlow outside the applications I've mentioned above, I would love to hear from you! Let me know in the comments section below this article and we'll discuss.

Installing TensorFlow is also a pretty straightforward task.

For CPU-only:

```
pip install tensorflow
```

For CUDA-enabled GPU cards:

```
pip install tensorflow-gpu
```

Learn how to build a neural network model using TensorFlow from the below comprehensive tutorials:

- An Introduction to Implementing Neural Networks using TensorFlow
- TensorFlow tutorials

Keras

Are you comfortable using Python? If yes, then you'll instantly connect with Keras. It is the perfect framework for you to start your deep learning journey.

Keras is written in Python and can run on top of TensorFlow (as well as CNTK and Theano). The TensorFlow interface can be a bit challenging as it is a low-level library and new users might find it difficult to understand certain implementations.

Keras, on the other hand, is a high-level API, developed with a focus to enable fast experimentation. So if want quick results, Keras will automatically take care of the core tasks and generate the output. Both Convolutional Neural Networks and Recurrent Neural Networks are supported by Keras. It runs seamlessly on CPUs as well as GPUs.

A common complaint from deep learning beginners is that they are unable to properly understand complex models. If you're one such user, Keras is for you! It is designed to minimize user actions and makes it really easy to understand models.

We can broadly classify models in Keras into two categories:

- **Sequential:** The layers of the model are defined in a sequential manner. This means that when we're training our deep learning model, these layers are implemented sequentially. Here is a sample example of a sequential model:

```
from keras.models import Sequential
from keras.layers import Dense


model = Sequential()
# we can add multiple layers to the model using .add()
```

```
model.add(Dense(units=64, activation='relu', input_dim=100))
model.add(Dense(units=10, activation='softmax'))
```

- **Keras functional API:** This is generally used for defining complex models, such as multi-output models or models with shared layers. Check out the below code to understand this in a practical manner:

```
from keras.layers import Input, Dense
from keras.models import Model

inputs = Input(shape=(100,)) # specify the input shape
x = Dense(64, activation='relu')(inputs)
predictions = Dense(10, activation='softmax')(x)

model = Model(inputs=inputs, outputs=predictions)
```

Keras has multiple architectures, mentioned below, for solving a wide variety of problems. This includes one of my all-time favorites – image classification!

1. VGG16
2. VGG19
3. InceptionV3
4. Mobilenet, and many more

You can refer to the official Keras documentation to get a detailed understanding of how the framework works.

You can install Keras with just one line of code:

```
pip install keras
```

Intrigued by Keras? Continue your learning with the below tutorial, where you'll understand how to implement a neural network using Keras:

- Optimizing Neural Networks using Keras

PyTorch

Remember when we said TensorFlow is the most commonly used deep learning framework right now? It might not hold that mantle for too long given the rapid pace with which data scientists and developers are embracing Facebook's PyTorch.

I am a huge PyTorch advocate. Among all the frameworks I have worked on, PyTorch is the most flexible.

PyTorch is a port to the Torch deep learning framework which can be used for building deep neural networks and executing tensor computations. Torch is a Lua-based framework whereas PyTorch runs on Python.

PyTorch is a Python package which provides Tensor computations. Tensors are multidimensional arrays just like numpy's ndarrays which can run on GPU as well. PyTorch uses dynamic computation graphs. Autograd package of PyTorch builds computation graphs from tensors and automatically computes gradients.

Instead of predefined graphs with specific functionalities, PyTorch provides a framework for us to build computational graphs as we go, and even change them during runtime. This is valuable for situations where we don't know how much memory is going to be required for creating a neural network.

You can work on all sorts of deep learning challenges using PyTorch, including:

1. Images (Detection, Classification, etc.)
2. Text (NLP)
3. Reinforcement Learning

If you're wondering how to install PyTorch on your machine, hold on for a moment. The installation steps vary depending on your operating system, the package you want to use to install PyTorch, the tool/language you're working with, CUDA and a few other dependencies.

Check the installation steps of PyTorch for your machine here. Once you're ready with the framework, check out the below two resources to build your first neural network using PyTorch:

- Learn How to Build Quick & Accurate Neural Networks using PyTorch – 4 Awesome Case Studies
- PyTorch tutorials

Caffe

Caffe is another popular deep learning framework geared towards the image processing field. It was developed by Yangqing Jia during his Ph.D at the University of Claifornia, Berkeley. And yes, it's open source as well!

First, a caveat – Caffe's support for recurrent networks and language modeling is not as great as the above three frameworks. But where Caffe stands out is it's speed of processing and learning from images. That is easily it's primary USP.

> *Caffe can process over sixty million images on a daily basis with a single NVIDIA K40 GPU. That's 1 ms/image for inference and 4 ms/image for learning.*

It provides solid support for interfaces like C, C++, Python, MATLAB as well as the traditional command line.

The Caffe Model Zoo framework allows us to access pretrained networks, models and weights that can be applied to solve deep learning problems. These models work on the below tasks:

1. Simple regression
2. Large-scale visual classification
3. Siamese networks for image similarity
4. Speech and robotics applications

You can check the installation of caffe and the documentation for more details.

Deeplearning4j

Any Java programmers in our community? Here's your ideal deep learning framework! Deeplearning4j is implemented in Java and is hence more efficient as compared to Python. It uses the tensor library called ND4J which provides an ability to work with n-dimensional arrays (also called tensors). This framework also supports both CPUs and GPUs.

Deeplearning4j treats the task of loading data and training algorithms as separate processes. This separation of functions provides a whole lot of flexibility. And who wouldn't like that, especially in deep learning?!

Deeplearning4j works with different data types as well:

1. Images
2. csv
3. plain text, etc.

The kind of deep learning models you can build using Deeplearning4j are:

1. Convolutional Neural Networks (CNNs)
2. Recurrent Neural Networks (RNNs)
3. Long Short-Term Memory (LSTM) and many other architectures.

Go through the installation steps and documentation of Deeplearning4j to get started with this framework.

Comparing these 5 Deep Learning Frameworks

We have covered five of the most popular deep learning frameworks out there. Each has it's own unique set of features which is why data scientists go for one over the other.

Have you decided which one you want to use? Or perhaps you are planning to switch to a completely new framework? Whatever the case, it's important to understand the advantages as well as limitations of each framework. You don't want to end up surprised if you face an error at some point!

Some frameworks work extremely well with image data but fail to parse through text data. Other frameworks perform well with both image and text data but their inner working can be difficult to understand.

In this section, we will compare our five deep learning frameworks using the below criteria:

1. Community support
2. Language in which they are written
3. Interface
4. Support for pretrained models

The table below compares these frameworks:

Deep Learning Framework	Release Year	Written in which language?	CUDA supported?	Does it have pretrained models?
TensorFlow	2015	C++, Python	Yes	Yes
Keras	2015	Python	Yes	Yes
PyTorch	2016	Python, C	Yes	Yes
Caffe	2013	C++	Yes	Yes

Deeplearning4j	2014	C++, Java	Yes	Yes

It's a pretty handy table for the next time you're working with these frameworks!

All of these frameworks are open source, support CUDA and have pretrained models to help you get started. But, what should be the right starting point and which framework should you choose to build your (initial) deep learning models? Let's discuss!

TensorFlow

We'll start with TensorFlow. TensorFlow works well on images as well as sequence-based data. If you are a beginner in deep learning, or don't have a solid understanding of mathematical concepts like linear algebra and calculus, then the steep learning curve of TensorFlow might be daunting.

I totally understand that this aspect can be complex for folks who are just starting out. My suggestion would be to keep practicing, keep exploring the community, and keep reading articles to get the hang of TensorFlow. Once you have a good understanding of the framework, implementing deep learning models will be very easy for you.

Keras

Keras is a pretty solid framework to start your deep learning journey. If you are familiar with Python and are not doing some high-level research or developing some special kind of neural network, Keras is for you.

The focus is more on achieving results rather than getting bogged down by the model intricacies. So if you are given a project related to, say image classification or sequence models, start with Keras. You will be able to get a working model very quickly.

Keras is also integrated in TensorFlow and hence you can also build your model using *tf.keras*.

PyTorch

As compared to TensorFlow, PyTorch is more intuitive. One quick project with both these frameworks will make that abundantly clear.

Even if you don't have a solid mathematics or a pure machine learning background, you will be able to understand PyTorch models. You can define or manipulate the graph as the model proceeds which makes PyTorch more intuitive.

PyTorch does not have any visualization tool like TensorBoard but you can always use a library like *matplotlib*. I wouldn't say PyTorch is *better* than TensorFlow, but both these deep learning frameworks are incredibly useful.

Caffe

Caffe works very well when we're building deep learning models on image data. But when it comes to recurrent neural networks and language models, Caffe lags behind the other frameworks we have discussed. The key advantage of Caffe is that even if you do not have strong machine learning or calculus knowledge, you can build deep learning models.

Caffe is primarly used for building and deploying deep learning models for mobile phones and other computationally constrained platforms.

Deeplearning4j

Like I mentioned before, Deeplearning4j is a paradise for Java programmers. It offers massive support for different neural networks like CNNs, RNNs and LSTMs. It can process a huge amount of data without sacrificing speed. Sounds like too good an opportunity to pass up!

End Notes & Illustrated Infographic

Are there any other deep learning frameworks you've worked on? I would love to hear your thoughts and feedback on that plus the ones we covered in this article. Connect with me in the comments section below.

And remember, these frameworks are essentially just tools that help us get to the end goal. Choosing them wisely can reduce a lot of effort and time.

As promised, here is the infographic with detailed about each deep learning framework we have covered. Download it, print it, and use it next time you're building a deep learning model!

TOP FIVE

DEEP LEARNING FRAMEWORKS

Deep learning frameworks are ubiquitous these days. Which one should you use? What are the advantages of one framework over the other? Let's find out below!

TENSORFLOW

- Developed by : Google Brain team
- TensorBoard for effective data visualization
- Written in C++ and Python
- Need more programming experience
- Static computation graphs
- Has a larger community base
- Plethora of learning resources

KERAS

- Developed by : François Chollet, Google engineer
- High level library and hence it enables fast experimentation
- Uses TensorFlow at the backend
- Easiest framework to start your deep learning journey

PYTORCH

- Developed by : Facebook's AI research group
- Dynamic Computation graphs
- Easy to use as compared to TensorFlow
- Written in C and Python
- More popular amongst researchers
- Not regularly used as a

- Not regularly used as a production framework

- Developed by : Berkeley AI Research (BAIR)
- Very quick image processing
- Not effective for RNNs and language models
- Used for deploying models for smart devices
- Written in C++

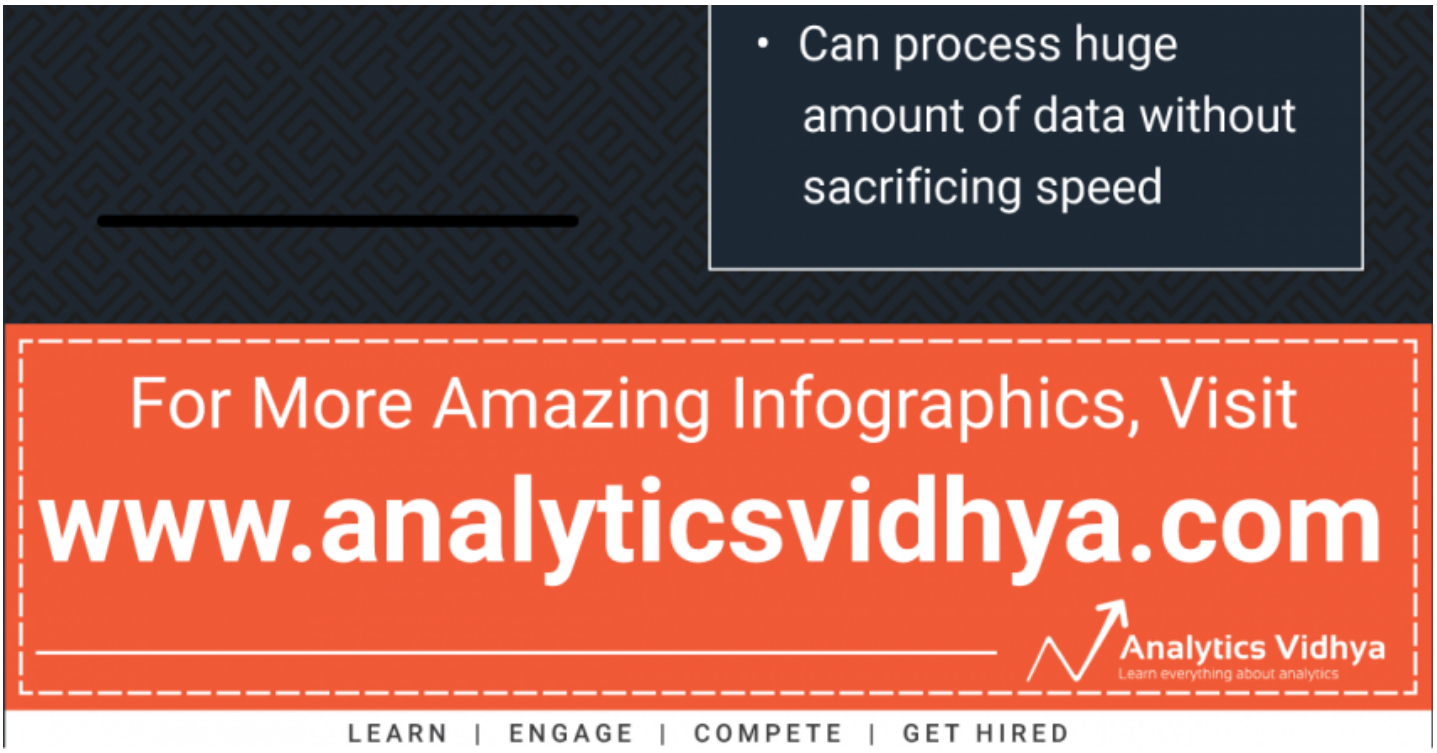
CAFFE

Caffe

DEEPLEARNING4J

DEEPLEARNING4J

- Developed by : Adam Gibson, Skymind
- Java library for deep learning
- Takes advantage of distributed frameworks (Spark/Hadoop)



You can also read this article on Analytics Vidhya's Android APP 

Share this:

Related Articles

[11 most read Deep Learning Articles from Analytics Vidhya in 2017](#)
December 21, 2017
In "Deep Learning"

[The 25 Best Data Science and Machine Learning GitHub Repositories from 2018](#)
December 26, 2018
In "Computer Vision"

[Tutorial on Text Classification (NLP) using ULMFiT and fastai Library in Python](#)
November 29, 2018
In "Deep Learning"

TAGS : CAFFE, DEEP LEARNING, DEEP LEARNING FRAMEWORK, DEEPLEARNING4J, KERAS, PYTHON, PYTORCH, TENSORFLOW

NEXT ARTICLE

8 Excellent Pretrained Models to get you Started with Natural Language Processing (NLP)

PREVIOUS ARTICLE

A Step-by-Step NLP Guide to Learn ELMo for Extracting Features from Text

Pulkit Sharma

My research interests lies in the field of Machine Learning and Deep Learning. Possess an enthusiasm for learning new skills and technologies.

8 COMMENTS

COSMIN CATALIN SANDA

March 14, 2019 at 1:55 pm

Reply

This list should probably have contained Apache MXNet as well, the production ready deep learning framework.

PULKIT SHARMA

March 15, 2019 at 11:37 am

Reply

Hi Cosmin,
Thanks for your feedback! Surely there are many other deep learning frameworks but for the scope of this article, I have only included the 5 most commonly used frameworks. If you have some points related to the Apache MXNet framework, please share them here, it will be helpful for the community.

KISHNA

March 14, 2019 at 2:21 pm

Reply

Thank you sir,
It is so helpful for me.

PULKIT SHARMA

March 15, 2019 at 11:33 am

Reply

Glad you liked it Kishna!

SATISH

March 15, 2019 at 4:38 pm

Reply

Truly a very good article on how to handle the future technology. After reading your post,thanks for taking the time to discuss this, I feel happy about and I love learning more about this topic.

ANKUSH

March 15, 2019 at 4:39 pm

Reply

Thank you sir,
It is so helpful for me. good keep it up

MARCEL

March 18, 2019 at 7:52 pm

Reply

Thanks for your nice article and your famous website.
Recently I've been studying fast.ai, https://www.fast.ai/
This is for Pytorch what Keras is for Tensorflow, but easier, with less code and even better results.
It's amazing beyond anything I've seen before.
Jeremy Howard, the inventor of fast.ai, calls it deeplearning 2018 as TF is deeplearning 2017.

PULKIT SHARMA

March 20, 2019 at 1:20 pm

Reply

Hi Marcel,
Thanks for this extremely helpful information. It will surely be helpful for the community.

LEAVE A REPLY

Your email address will not be published.

Comment

Name (required)

Email (required)

Website

SUBMIT COMMENT

Notify me of new posts by email.

JOIN THE NEXTGEN DATA SCIENCE ECOSYSTEM

- Get access to free courses on Analytics Vidhya
- Get free downloadable resource from Analytics Vidhya
- Save your articles
- Participate in hackathons and win prizes

Join Now

POPULAR POSTS

- 6 Useful Programming Languages for Data Science You Should Learn (that are not R and Python)
- 24 Ultimate Data Science Projects To Boost Your Knowledge and Skills (& can be accessed freely)
- Commonly used Machine Learning Algorithms (with Python and R Codes)
- 7 Regression Techniques you should know!
- A Complete Python Tutorial to Learn Data Science from Scratch
- Stock Prices Prediction Using Machine Learning and Deep Learning Techniques (with Python codes)
- Understanding Support Vector Machine algorithm from examples (along with code)
- Introduction to k-Nearest Neighbors: A powerful Machine Learning Algorithm(with implementation in Python & R)

RECENT POSTS

10 Artificial Intelligence (AI) Startups in India You Should Know

JULY 2, 2019

6 Powerful Open Source Machine Learning GitHub Repositories for Data Scientists

JULY 1, 2019

Top 5 Must-Read Answers – What does a Data Scientist do on a Daily Basis?

JUNE 27, 2019

Announcing DataHack Summit 2019 – The Biggest Artificial Intelligence and Machine Learning Conference Yet

JUNE 26, 2019