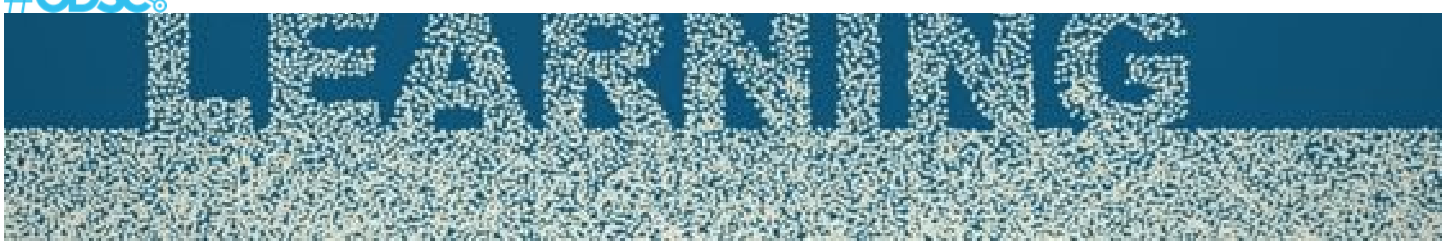
Deep Learning Part 1: Comparison of Symbolic Deep Learning Frameworks

#ODSC (https://www.opendatascience.com)



By Anusua Trivedi, Sr. Data Scientist - Microsoft | 10/17/2016

Tags: Deep Learning (https://www.opendatascience.com/tag/deep-learning/)

Background and Approach

This blog series is based on my upcoming talk on re-usability of Deep Learning Models at the Hadoop+Strata World Conference (http://conferences.oreilly.com/strata/hadoop-big-data-sg/public/schedule/detail/54278) in Singapore. This blog series will be in several parts – where I describe my experiences and go deep into the reasons behind my choices.

Deep learning is an emerging field of research, which has its application across multiple domains. I try to show how transfer learning and fine tuning strategy leads to re-usability of the same Convolution Neural Network model in different disjoint domains. Application of this model across various different domains brings value to using this fine-tuned model.

In this blog (Part1), I describe and compare the commonly used open-source deep learning frameworks. I dive deep into different pros and cons for each framework, and discuss why I chose Theano (https://github.com/Theano/Theano) for my work.

Symbolic Frameworks

Symbolic computation frameworks (as in CNTK (https://github.com/Microsoft/CNTK), MXNET (https://github.com/dmlc/mxnet), TensorFlow (https://www.tensorflow.org/), Theano) are specified as a symbolic graph of vector operations, such as matrix add/multiply or convolution. A layer is just a composition of those operations. The fine granularity of the building blocks (operations) allows users to invent new complex layer types without implementing them in a low-level language (as in Caffe (http://caffe.berkeleyvision.org/)).

I've used different symbolic computation frameworks in my work. However, I found each of them has their pros and cons in their design and current implementation, and none of them can perfectly satisfy all needs. For my problem needs , I decided to work with Theano.

Here we compare the following symbolic computation frameworks:

Theano (https://github.com/Theano/Theano)

- Software: Theano
- Creator: Université de Montréal
- Software license: BSD license
- Open source: Yes
- Platform: Cross-platform
- Written in: Python
- Interface: Python
- CUDA support: Yes
- Automatic differentiation: Yes
- Has pre-trained models: Through Lasagne's model zoo
- Recurrent Nets: Yes
- Convolutional Nets: Yes
- RBM/DBNs: Yes

TensorFlow
(https://github.com/tensorflow/tensorflow/issues/654)%20(which%20is%20used%20frequently%20in%20sequence%20prediction%20tasks)

- Software: TensorFlow
- Creator: Google Brain Team

- Open source: Yes
- Platform: Linux, Mac OS X,
- Windows support on roadmap
- Written in: C++, Python
- Interface: Python, C/C++
- CUDA support: Yes
- Automatic differentiation: Yes
- Has pre-trained models: No
- Recurrent Nets: Yes
- Convolutional Nets: Yes
- RBM/DBNs: Yes

MXNET (https://github.com/dmlc/mxnet/)

- Software: MXNET
- Creator: Distributed (Deep) Machine Learning Community
- Software license: Apache 2.0
- Open source: Yes
- Platform: Ubuntu, OS X, Windows, AWS, Android, iOS, JavaScript
- Written in: C++, Python, Julia, Matlab, R, Scala
- Interface: C++, Python, Julia, Matlab, JavaScript, R, Scala
- CUDA support: Yes
- Automatic differentiation: Yes
- Has pre-trained models: Yes
- Recurrent Nets: Yes

- Convolutional Nets: Yes
- RBM/DBNs: Yes

Non-symbolic frameworks

PROS:

- Non-symbolic (imperative) neural network frameworks like ***torch*** (https://github.com/torch/), ***caffe*** (https://github.com/BVLC/caffe/) etc. tend to have very similar design in their computation part.
- In terms of expressiveness, imperative frameworks with a good design can also expose graph-like interface (e.g. ***torch/nngraph*** (https://github.com/torch/nngraph)).

CONS:

- The main drawbacks of imperative frameworks actually lie in manual optimization. For example, in-place operation has to be manually implemented.
- Most imperative frameworks are not designed well enough to have comparable expressiveness as symbolic frameworks.

Symbolic frameworks

PROS:

- Symbolic frameworks can possibly infer optimization automatically from the dependency graph.
- A symbolic framework can exploit much more memory reuse opportunities, as is well done in ***MXNET***.
- Symbolic frameworks can automatically compute an optimal schedule. This is explained in ***TensorFlow whitepaper*** (http://download.tensorflow.org/paper/whitepaper2015.pdf).

CONS:

(https://www.opendatascience.com)

Adding New Operations

In all of these frameworks, adding an Operation with reasonable performance is not easy.

Theano / MXNET	**TensorFlow**
Can add Operation in Python with inline C support.	Forward in C++, symbolic gradient in Python.

Code Re-usability

Training deep networks are time-consuming. So, Caffe has released some pre-trained model/weights (model zoo) which could be used as initial weights while transfer learning or fine tuning deep networks on domain specific or custom images.

- **Theano**
  Lasagne is a high-level framework built on top of Theano. It's very easy to use Caffe pre-tained model weights in Lasagne.
- **TensorFlow**
  No support for pre-trained model.
- **MXNET**
  MXNET has a *caffe_converter tool* (https://github.com/dmlc/mxnet/tree/master/tools/caffe_converter) which allows to convert pre-trained caffe model weights to fit MXNET.

Low-level Tensor Operators

A reasonably efficient implementation of low-level operators can serve as ingredients in writing new models, saving the effort to write new Operations.

Theano	**TensorFlow**	**MXNET**
A lot of basic Operations	Fairly good	Very few

Control Flow Operator

Control flow operators make the symbolic engine more expressive and generic.

Theano	TensorFlow	MXNET
Supported	Experimental	Not Supported

High–level Support

- **Theano**
  Pure symbolic computation framework. High-level frameworks can be built to fit desired means of use. Successful examples include *Keras* (http://keras.io/), *Lasagne* (http://lasagne.readthedocs.org/en/latest/), *blocks* (http://blocks.readthedocs.org/en/latest/).
- **TensorFlow**
  Has good design considerations for neural network training, and at the same time avoid being totally a neural network framework, which is a wonderful job. The***graph collection***, ***queues***, ***image augmenters*** etc. can be useful building blocks for a higher-level wrapper.
- **MXNET**
  Apart from the symbolic part, MXNET also comes with all necessary *components* (https://github.com/dmlc/mxnet/tree/master/example/image-classification) for image classification, going all the way through data loading to building a model that has a method to start training.

Performance

Benchmarking Using Single-GPU

I benchmark LeNet model on MNIST Dataset using a Single-GPU (NVIDIA Quadro K1200 GPU).

Theano	TensorFlow	MXNET
Great	Not so good	Excellent

(https://www.opendatascience.com)

Memory

GPU memory is limited and may usually be a problem for large models.

Theano	TensorFlow	MXNET
Great	Not so good	Excellent

Single–GPU Speed

Theano takes a long time to compile a graph, especially with complex models. **TensorFlow** is a bit slower.

Theano / MXNET	TensorFlow
comparable to CuDNNv4	about 0.5x slower

Parallel/Distributed Support

Theano	TensorFlow	MXNET
experimental multi-GPU	multi-GPU	distributed

Conclusion

Theano (with higher-level Lasagne & Keras) is a great choice for deep learning models. It's very easy to implement new networks & modify existing networks using Lasagne/Keras. I prefer python, and thus prefer using Lasagne/Keras due to their very mature python interface. However, they do not support R. I have tried using transfer learning and fine tuning in Lasagne/Keras, and it's very easy to modify an existing network and customize it with domain-specific custom data.

Comparisons of different frameworks show that MXNET is the best choice (better performance/memory). Moreover, it has a great R support. In fact, it is the only framework that supports all functions in R (http://dmlc.ml/rstats/2015/11/03/training-deep-net-with-R.html). In MXNET, transfer learning and fine tuning networks are possible, but not as easy (as compared to Lasagne/Keras). This makes modifying existing trained networks more difficult, and thus a bit difficult to use domain-specific custom data.

Originally posted at blog.revolutionanalytics.com (http://blog.revolutionanalytics.com/2016/08/deep-learning-part-1.html). Please feel free to email Anusua at *trivedianusua23@gmail.com* if you have questions.

WHAT IS OPEN DATA SCIENCE?

Username *

Email *

Password *

Repeat Password *

Get Newsletter & Full Access

LATEST POSTS

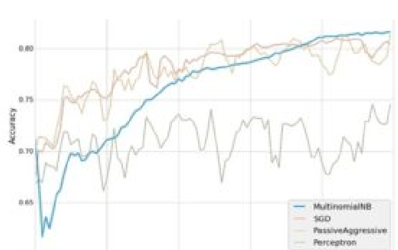
Text Mining Trump Speeches Part 3: Topic Modeling, Trump fixated on Hillary (https://www.opendatascience.com/blog/text-mining-trump-speeches-part-3-topic-modeling-trump-fixated-on-hillary/)
10/26/2016

#ODSC(https://www.opendatascience.com)

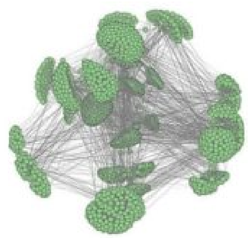
to-python/)
10/25/2016

Three Challenges for Open Data Science (https://www.opendatascience.com/conferences/three-challenges-for-open-data-science/)
10/25/2016

POPULAR POSTS

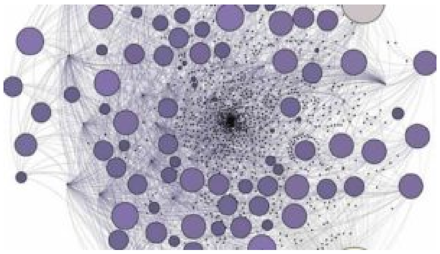


Riding on Large Data with Scikit-learn (https://www.opendatascience.com/blog/riding-on-large-data-with-scikit-learn/)
9494 views



R or Python for data science? (https://www.opendatascience.com/blog/r-or-python-for-data-science/)

9388 views



Data Science on Twitter (https://www.opendatascience.com/blog/data-science-on-twitter/)

5931 views

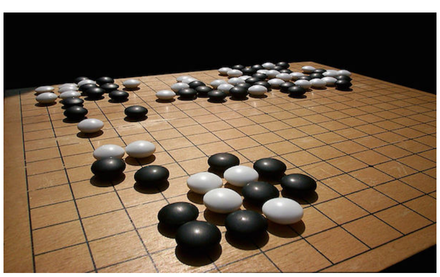
RELATED POSTS

What Combining 50,000 Fonts Using Neural Networks Looks Like (https://www.opendatascience.com/blog/what-combining-50000-fonts-using-neural-networks-looks-like/)

#ODSC(https://www.opendatascience.com)

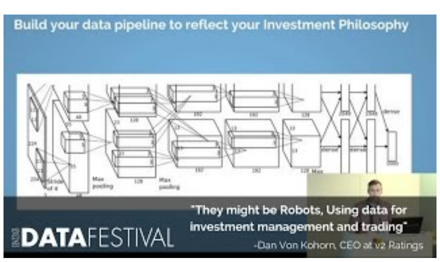
TensorFlow Release Takes Data Science Community by Storm (https://www.opendatascience.com/blog/tensorflow-release-takes-data-science-community-by-storm/)

12/14/2015



One Small Move on the Go Board, One Giant Move for AI (https://www.opendatascience.com/blog/one-small-move-on-the-go-board-one-giant-move-for-ai/)

01/29/2016



Dan Von Kohorn – "…Using Data for Investment Management & Trading" (https://www.opendatascience.com/conferences/dan-von-kohorn-at-bdf2015-they-might-be-robots-using-data-for-investment-management-and-trading/)

11/04/2015