Comparison and Analysis of the Open-Source Frameworks for Deep Learning

Dong-sheng GAO\textsuperscript{1,*}, Yan-rong ZHAO\textsuperscript{2}, Jing GAO\textsuperscript{3} and Hao WANG\textsuperscript{3}

\textsuperscript{1}State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, China
\textsuperscript{2}Hebei Geomatics Center, Hebei Shijiazhuang, China
\textsuperscript{3}31434 Troops, China

*Corresponding author

Keywords: Deep learning, Open-source frameworks, Performance comparison.

Abstract. Deep Learning is the hottest trend now in AI and Machine Learning. The paper introduces four mainstream open-source frameworks for deep learning, including Caffe, TensorFlow, CNTK, Torchnet. And the open-source frameworks for deep learning are analyzed and compared from the aspects of network and model capability, interface, model deployment, performance, cross-platform and distributed. Finally, four open-source frameworks for deep learning are simulated on the dataset, and the advantages and disadvantages of each framework and its applicability are summarized.

Introduction

Deep learning is a new field in machine learning research. Its motivation lies in the establishment and simulation of the neural network of human brain for analysis and learning, which imitates the mechanism of human brain to interpret data such as images, sounds and texts. The concept of deep learning is derived from the study of artificial neural networks. Multi-layer perceptron with multiple hidden layers is a kind of deep learning structure. Deep learning is a fashionable way of learning, which creates a more abstract high-level representation of attribute categories or characteristics by combining lower-level features to discover the distributed feature representation of the data.

As the deep learning research boom continues to ferment, there are a variety of the open-source frameworks for deep learning, which include Caffe\textsuperscript{[1]}, CNTK\textsuperscript{[2]}, TensorFlow\textsuperscript{[3]}, and Torchnet\textsuperscript{[4]}, etc. This paper does a full comparison of these deep learning frameworks, from the network, model capability, interface, deployment, performance and cross-platform.

Introduction of Frameworks

TensorFlow

TensorFlow is the second-generation artificial intelligence learning system developed by Google based on DistBelief. It uses for speech recognition\textsuperscript{[5]} or image recognition and other deep learning. It supports CNN\textsuperscript{[6]}, RNN\textsuperscript{[7]} and LSTM\textsuperscript{[8]} algorithms, and supports both C++ and python interfaces. TensorFlow 0.8 supports distributed computing, which can run on hundreds of servers.

Caffe

Caffe is the most widely used machine learning framework, and used to calculate the CNN-related algorithms, mainly for image recognition, which supports command line, Python and MATLAB interfaces, can be seamlessly switched on the CPU and GPU, does not support distributed computing.

Caffe uses a typical function (process) calculation. It defines the abstract class Layer, which receives Blob data and calculates the output of the new Blob, as the input to the upper layer of Layer. Finally, the different layers are combined into net.
**CNTK**

CNTK is Microsoft's the open-source toolkit for deep learning. Through the fine-grained building blocks, users can create new, complex layer types without requiring the use of low-level languages. CNTK supports FNN, KNN, LSTM, and FNN algorithms and distributed computing. It uses a command-line interface. That is, it runs the command line through the specified configuration file.

**Torchnet**

Torchnet is the Facebook’s deep learning development framework. Torchnet is fundamentally different from the deep learning frameworks such as Caffe, TensorFlow, which provides a framework (i.e. torch / nn) over the deep learning framework to facilitate rapid testing. Torchnet provides a set of subsets of packages and performs five major types of abstraction, and he most important subset of packages is responsible for computer vision, natural language processing and speech processing.

**Comparison of Frameworks**

**Networks and Modeling Capability**

**Caffe.** Caffe is perhaps the first mainstream industry-grade deep learning toolkit, started in late 2013, due to its excellent convolutional neural network implementation. It is still the most popular toolkit within the computer vision community, with many extensions being actively added. However, its support for recurrent networks and language modeling in general is poor, due to its legacy architecture. In addition, in Caffe, we define a layer with C++, while a network uses Protobuf.

**CNTK.** In CNTK, a network is 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.

**TensorFlow.** Since TensorFlow uses symbolic graph of vector operations approach, specifying a new network is fairly easy. TensorFlow does not support bidirectional RNN and 3D convolution, and the common version of the graph definition does not support looping and conditional control, which makes the implementation of RNN not ideal, because Python loop must be used and graph compilation optimization is impossible.

**Torchnet.** Torchnet supports the convolution networks. Its use is very intuitive for Torch's native interface of time-domain convolution. Torch supports a large number of RNNs through many unofficial extensions, and there are a number of ways to define networks. But Torch essentially defines the network as a layer, and this coarse-grained approach lacks sufficient support for new layer type extensions. Compared to Caffe, it is very easy to define new layers in Torch, without the need for C++ programming. The distinction between layer and network definition is minimal.

**Interfaces**

**Caffe.** Caffe has command line, Python interface and MATLAB interface.

**CNTK.** CNTK is to specify a configuration file and run command line.

**TensorFlow.** TensorFlow supports two interfaces: Python and C++. This means that you can do experiments in a rich, high-level environment and deploy your model in an environment that requires native code or low latency.

**Torchnet.** Torch runs on LuaJIT, which is amazingly fast (comparable with industrial languages such as C++/C#/Java). Hence they can just write all kinds of computations without worrying about performance penalty. However, Lua is not yet a mainstream language.
Model Deployment

**Caffe.** Caffe is C++ based, which can be compiled on a variety of devices. It is cross-platform, which makes Caffe the best choice with respect deployment.

**CNTK.** Like Caffe, CNTK is also C++ based and is cross-platform. Hence, deployment should be easy in most cases. However, to my understanding, it doesn't work on ARM architecture, which limits its capability on mobile devices.

**TensorFlow.** TensorFlow supports C++ interface and the library can be compiled/optimized on ARM architectures because it uses Eigen (instead of a BLAS library). This means that you can deploy your trained models on a variety of devices (servers or mobile devices) without having to implement a separate model decoder or load Python/LuaJIT interpreter. TensorFlow doesn't work on Windows yet so TF models can't be deployed on Windows devices though.

**Torchnet.** Torch require LuaJIT to run models. This makes it less attractive than bare bone C++ support of Caffe/CNTK/ TensorFlow. It’s not just the performance overhead, which is minimal. The bigger problem is integration, at API level, with a larger production pipeline.

Performance

In single-GPU, all of these toolkits call cuDNN so as long as there’s no major computations or memory allocations at the outer level, they should perform similarly.

**Caffe.** Caffe is simple and fast.

**CNTK.** CNTK is simple and fast.

**TensorFlow.** TensorFlow used to be slow at first, but now is equal with other platforms when version 0.8 came out at 05/2016.

**Torchnet.** Torchnet is very good, there is no problem with TensorFlow. Speed Comparison of these Open-Source Frameworks is illustrated in Figure 1.

**Note:** TensoFlow in the Figure 1 is the performance of the old version. Its performance of 0.8 version is comparable to Torch.

Cross-platform and Distributed

Caffe and CNTK work on all OSes. TensorFlow and Torch do not work on Windows.

CNTK and TensorFlow support distributed, Caffe and Torchnet only support CPU, single GPU, single multi-GPUs.

Analysis of Experimental Results

Database Description

CIFAR-10[9] is an image data set, containing 60,000 32 * 32 labeled color images, divided into 10 categories, each category 6000. The data set is divided into 6 batches, 5 training batches and 1 test batch. Each batch contains 10,000 images. The test batch consists of randomly selected 1000 images in each type of image, and the remaining images are randomly divided into 5 training batches.

The Figure 2 shows a sample of the data set, a total of 10 categories, randomly selected 10 images from each type.
Figure 1. Speed Comparison of Open-Source Frameworks.

Figure 2. Sample Dataset.

Figure 3. Diagram of the Network Model.

Description of Network Model

The network model used in the experiment is shown below:

The hyperparameter settings in the network are as follows:
(1) Base learning rate base_lr is 0.001. It is reduced once every 5000 iterations, the decline factor is 10. That is, learning rate is 0.0001 after 5000 iterations, and so on ......
(2) Momentum momentum: 0.9;
(3) Weight penalty factor weight_decay: 0.004;
(4) The maximum number of iterations max_iter: 10000;
(5) The batch size batch_size: 100, batch_size samples are selected for training in each iteration.
Experimental Results

<table>
<thead>
<tr>
<th>Frameworks</th>
<th>Accuracy</th>
<th>Training Time[min]</th>
</tr>
</thead>
<tbody>
<tr>
<td>TensorFlow</td>
<td>0.750</td>
<td>62</td>
</tr>
<tr>
<td>Caffe</td>
<td>0.764</td>
<td>90</td>
</tr>
<tr>
<td>CNTK</td>
<td>0.738</td>
<td>56</td>
</tr>
<tr>
<td>Torchnet</td>
<td>0.781</td>
<td>77</td>
</tr>
</tbody>
</table>

Conclusion and Outlook

Finally, the summary of the above four kinds of frameworks, the following table is the performances comparison of 4 kinds of frameworks.

<table>
<thead>
<tr>
<th>Frameworks</th>
<th>Development company</th>
<th>Development languages</th>
<th>Supported interfaces</th>
<th>Supported models</th>
<th>Cross-platform</th>
<th>Speed</th>
<th>Distributed computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>TensorFlow</td>
<td>Google</td>
<td>c++/cuda, python</td>
<td>c++/python</td>
<td>CNN/RNN LSTM</td>
<td>not support windows</td>
<td>fast</td>
<td>yes</td>
</tr>
<tr>
<td>Caffe</td>
<td>University of California</td>
<td>c++/cuda, python</td>
<td>c++/python, matlab</td>
<td>CNN</td>
<td>cross-platform</td>
<td>normal speed</td>
<td>no</td>
</tr>
<tr>
<td>CNTK</td>
<td>Microsoft</td>
<td>c++/cuda</td>
<td>CNN/RNN LSTM</td>
<td>not support ARM</td>
<td>fastest</td>
<td>fastest</td>
<td>yes</td>
</tr>
<tr>
<td>Torchnet</td>
<td>Facebook</td>
<td>Lua/c++</td>
<td>Lua</td>
<td>CNN/RNN LSTM</td>
<td>not support windows</td>
<td>fast</td>
<td>no</td>
</tr>
</tbody>
</table>

As can be seen from the table 2, TensorFlow and CNTK support distributed computing, if you want to develop distributed computing model, you can only use these two frameworks; TensorFlow does not support windows operating system, CNTK does not support ARM architecture, which limits its ability on mobile devices; Therefore, the need to choose the appropriate framework for the actual situation. Caffe only supports the CNN model and can only handle image classification; Torchnet runs very fast on LuaJIT, and users can write any type of computation without worrying about performance. And Torchnet provides researchers with clear guidelines on how to build code and boilerplate code that speeds development, but Lua is not the language of the mainstream; therefore, the users of Torchnet need to learn a new language.

On the development perspective, TensorFlow is an open-source framework for deep learning by Google. The performance of the latest 0.8 version is comparable with the current mainstream frameworks. TensorFlow supports distributed computing, can be compiled and optimized based on ARM architecture. The users can deploy the model to a variety of well-trained equipment. They have TensorFlow as their development tools of deep learning for many industrial users and university researchers. Torchnet is the deep learning development framework announced by Facebook that provides abstraction code and boilerplate code for machine learning experiments, which makes rapid testing possible. There is an active developer community that creates optimization packages, manifold learning, metric learning, and neural networks and more. So Torchnet become another mainstream deep learning framework, has been welcomed by many scholars.

References


