Get started with TensorFlow

2016-06-21
uoguelph-mlrg
He Ma

This introduction Not aiming to be thorough

I only choose to introduce certain aspects of TensorFlow features here.

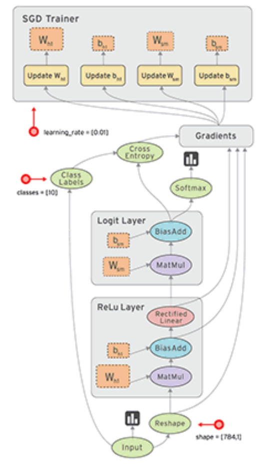
Also since I’m also new to tensor flow, I welcome any correction in explaining the basic concepts.

At a Glance

Basic Concepts in TensorFlow:

- graph
- edge
- node
  - operation
  - placeholder
  - variable

- tensor
- session



credit: https://www.tensorflow.org

Here on the right side shows a tensor flow graph. Like in theano, a computation graph describes mathematical computation to be carried out. It consists of edges and nodes. Edges are the connections between nodes showing the directions of data flow.

Nodes are the actual carrier of those internal computation and final result. A node can carry an operation, a placeholder or a variable.

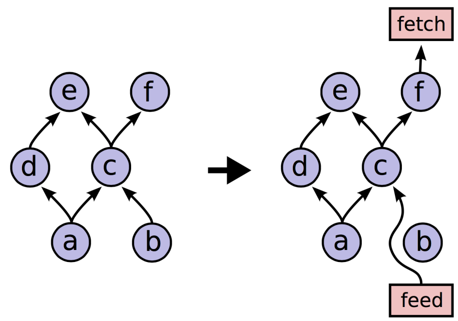
tensors are the implementation of those nodes in computer: dynamically-sized multidimensional data arrays

Session is a special term in TensorFlow. It is responsible for managing the computation graph. Typically one session manages one graph. Session.run() is the function call which starts the computation of a graph.

Nodes are assigned to computational devices and execute asynchronously and in parallel once all the tensors on their incoming edges become available.

Special Features

- Auto differentiation & partial execution



[1] Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems

TensorFlow also provides auto differentiation like theano.

Partial execution reduces redundant computation in a graph

Special Features

- Auto differentiation & partial execution
- Sync.and async. data parallelism
- Node to device placement & cross device comm.
- Graph visualization (TensorBoard)
- Performance tracing (EEG)

[1] Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems

In the recent release of TF, it starts to support distributed training.

Predecessor in Google Brain project: “Distbelief”

Performance tracing is a technique that is not open sourced yet. It helps tensor flow developers find out where is the performance bottleneck during graph execution.

Strength

- Flexiblility for quick idea and research
- High performance for production
- Portability: wide devices and platforms support
- Good documentation and wide user community

[1] Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems

TensorFlow also provide you high performance once your designed algorithm is mature and ready to be put into production.

It provides low level tensor operations like theano.

Wide device and platform support: CPUs or GPUs, and on desktop, server, or mobile computing platforms and cloud computing

Good Doc as well as a Wide user community:
most users and companies who were using Distbelief, the predecessor of TF, has switched to TensorFlow.
Abundant user feed back: bug report, enhancement from Github

Getting Started on copper

- Installation: Tools
  1. load installation dependencies
  2. install or upgrade your pip
  3. use pip to install TensorFlow0.9
- Run:
  4. Source dependencies (set4tf.sh)

After talking about its features. Let's try it out on copper.

1. If you are using a mac, it is very easy to install. However, if your model is large and you need GPU support, you probably want to let those computation done on a GPU cluster.
2. installation guide on mosaic and copper (Fei Mao)
3. run it on mosaic and copper

Examples

- Fit a line
- Run MNIST training

After installation, lets look at two examples on how to program in TensorFlow.

Two examples
can be multi-GPUs

Four Phases

1. create a model,
2. feed the input data,
3. run the model,
4. process/print/visualize the output.

I showed how to create a model, how to …, how to …

TensorBoard

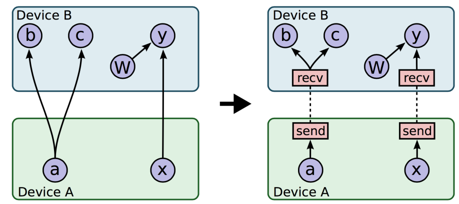
- Visualization:
  - Computation Graphs
  - Summary Data (scalar, histogram, image)
- Example:
  - visualize mnist training (http://0.0.0.0:6006)

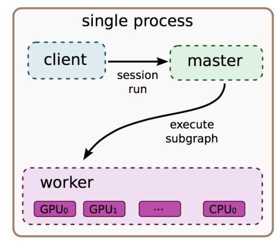
visualize the cifar10 model

visual check on the graph connection

Distributed Execution

- Multi-Device Execution
  - Node Placement
  - Cross Device Comm.



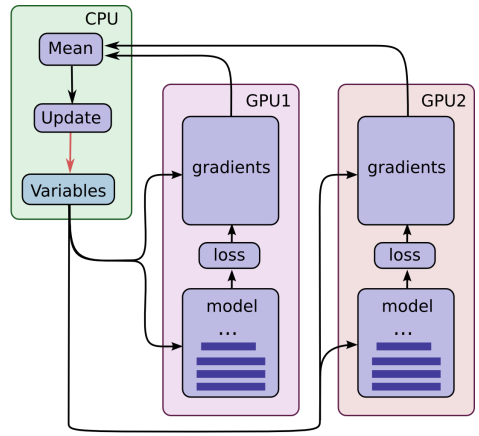


[1] Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems

Those following slides are probably more interesting to hardware acceleration researchers.

Distributed Execution

- Data Parallelism
  - synchronous
  - asynchronous



[1] Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems

Similarity to Theano-MPI:

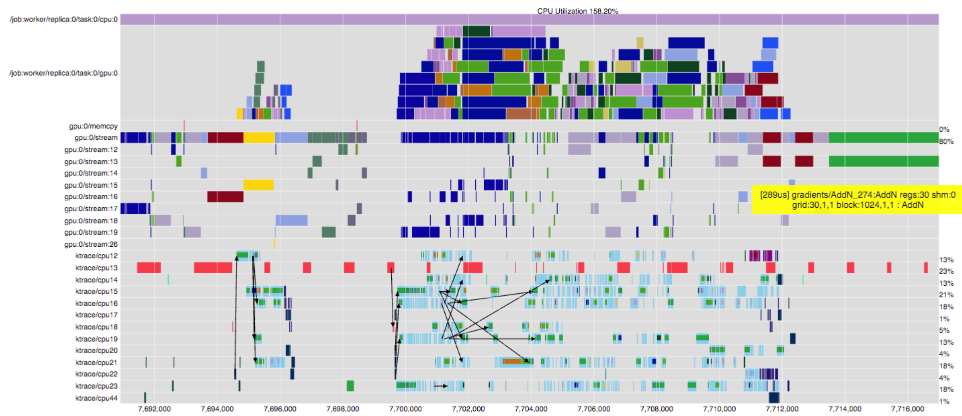
1. They sum gradients as well.
2. They do synchronous as well.

The difference here is that they move the updating part onto CPU, so there’s frequent data transfer device and host.

3. They used constant effective batch size as well.
4. They found out the problem with async as well.

Performance Tracing (EEG)

- Runtime profiling (Not open-sourced so far)



[1] Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems

It is for visualizing the exact ordering and performance characteristics of the execution of TensorFlow graphs:

light blue: host threads that enqueues GPU operations
other color: Housekeeping threads
arrows: threads stalled on GPU-CPU tranfers or queueing delay

1. execution of different kernels on different devices

2. data transfer between device and host and between CPUs
We can also kind of see the result of subgraph placement

Some Optimizations

- Scheduling of operations and node placement
- Controlling memory usage and network contention
- Lossy compression of comm. between devices

[1] Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems

1. Scheduling the order of execution so that more node can be run in parallel.
2. Controlling the maximum number of operations can be executed to control memory usage.
3. Controlling data comm to reduce network contention.
4. They taking advantage of low precision comm. as well as lossy comm. within machine and across machine boundaries.

This is possible because
"Some machine learning algorithms, including those typically used for training neural networks, are tolerant of noise and reduced precision arithmetic.
"

Recommended Reading

- Experience on using TensorFlow: Paper
- TensorFlow tutorials
- TensorFlow examples
- Wrappers of TensorFlow: Keras, PrettyTensor

Finally I suggest some further reading if anyone is interested in learning more about TensorFlow.

1. Sec6
2. Some comments on TF and comparison of TF with other frameworks.
3. TensorFlow Tutorials
4. TensorFlow examples
5. High Level wrappers of TF include Keras and Pretty Tensor

Thank you