Tensorflow - A system for large-scale machine learning

Presentation: Nat McAleese (nm583)

Structure

An introduction to the problem domain

Previous work

An explanation of Tensorflow

Results

Critique

Very brief introduction to neural networks

Smooth function optimisation.

An *iterative* optimisation procedure.

batch SGD - note that very large batches are *worse*

Not ‘embarrassingly parallel’

What is the problem?

Training large models requires a great deal of both data and compute. Thus it is important to be efficient and distributed [0, etc]

Progress in ML is empirically driven - architectures change frequently; results can be counter-intuitive. This necessitates flexible systems for rapid experimentation.

Examples: Hogwild [1], Async replication [2], Sync replication[3].

[0] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., & Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538;
[1] Recht, B., Re, C., Wright, S., & Niu, F. (2011). Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In Advances in neural information processing systems (pp. 693-701).
[2] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., ... & Ng, A. Y. (2012). Large scale distributed deep networks. In Advances in neural information processing systems (pp. 1223-1231).
[3] Chen, J., Monga, R., Bengio, S., & Jozefowicz, R. (2016). Revisiting distributed synchronous SGD. arXiv preprint arXiv:1604.00981.

What is the problem (with existing solutions)?

“Parameter Server” architectures become inefficient as more complexity is introduced into the update rule of the gradient descent algorithm [0].

Distributed deep learning systems were quite inflexible - layer-level, not operation-level design. [1, 2]

Theano was single machine only. [3]

Other dataflow designs were not efficient under the relaxed consistency requirements of ML.

[0] Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
[1] Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., ... & Zhang, Z. (2015). Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274.
[2] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., ... & Ng, A. Y. (2012). Large scale distributed deep networks. In Advances in neural information processing systems (pp. 1223-1231).
[3] (many authors). Theano: A Python framework for fast computation of mathematical expressions. arXiv preprint, 1605.02688, 2016. arxiv.org/abs/1605.02688.
[4] “Spark takes 20 seconds to broadcast weights and collect updates from five workers...” - See the Tensorflow paper.

What is Tensorflow?

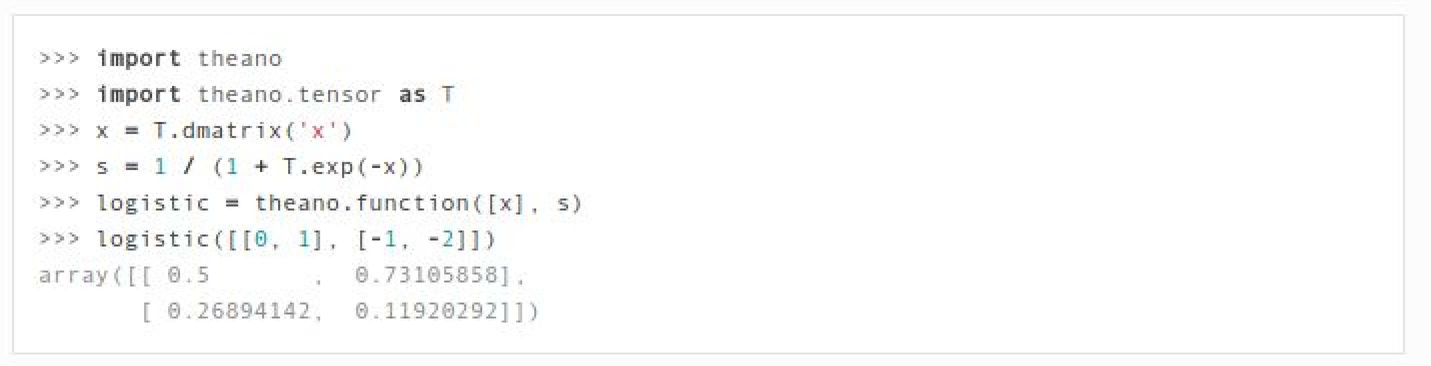
Distributed Theano?

Theano + Dryad?

What was Theano?

“Theano is a Python library that allows to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently”

Never considered distributed computing a primary goal.



```
>>> import theano
>>> import theano.tensor as T
>>> x = T.dmatrix('x')
>>> s = 1 / (1 + T.exp(-x))
>>> logistic = theano.function([x], s)
>>> logistic([[0, 1], [-1, -2]])
array([[ 0.5       ,  0.73105858],
       [ 0.26894142,  0.11920292]])
```

Note the fine-grained control unavailable in say, Caffe, due to the op-level API

Source: http://deeplearning.net/software/theano/tutorial/examples.html

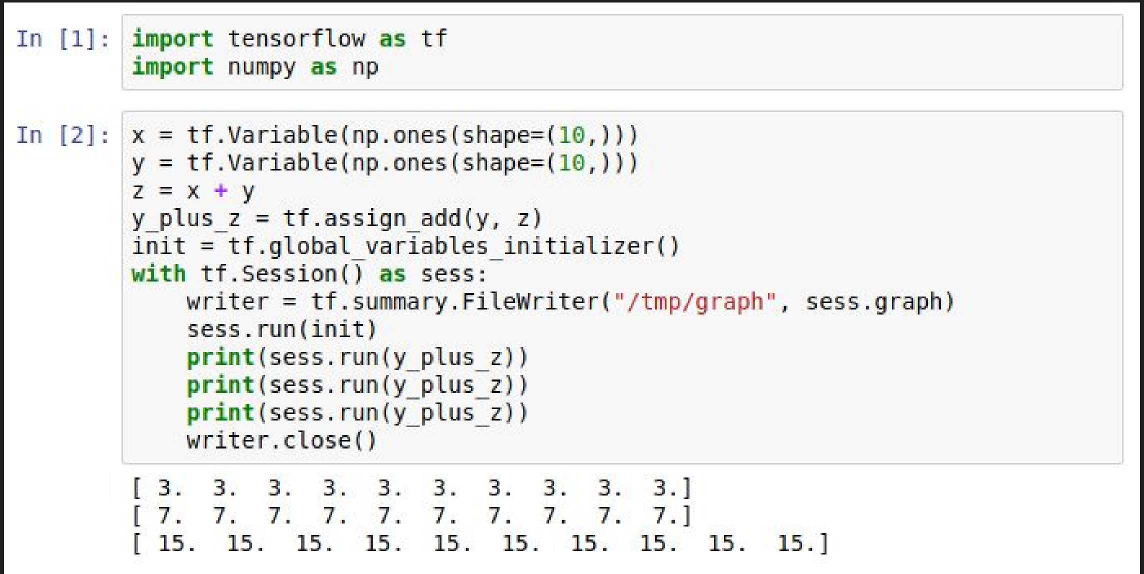
What was Theano?

Two types of node: Variables and Apply Nodes (including Scan, which is a little special)

Two steps: graph compilation and execution.

This limited programming model allows for simple automatic differentiation, many algebraic graph optimisations to improve both performance and numerical stability, as well as specific compilation for available hardware - *such as GPUs*.

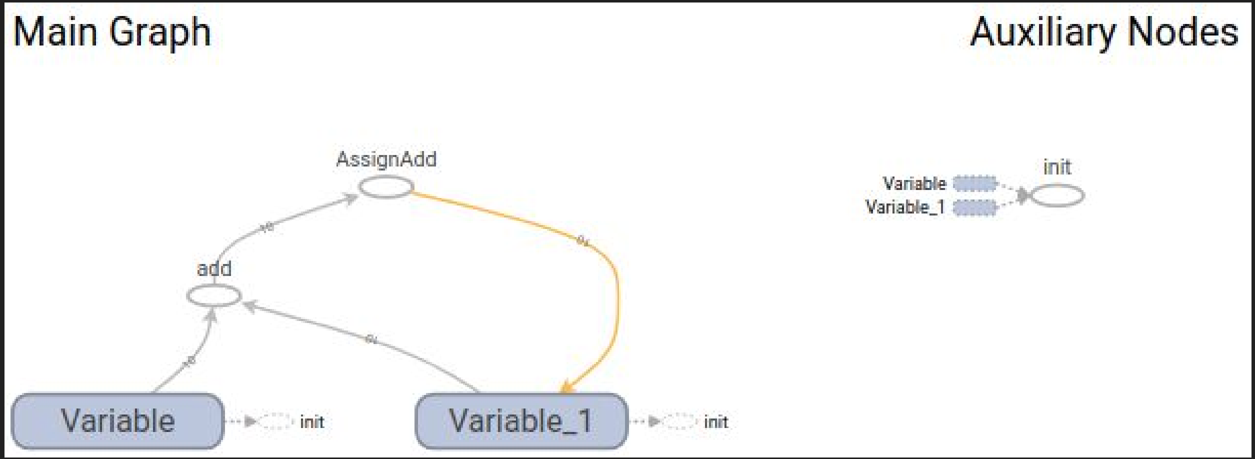
It also allows for automatic parallelization, but we'll discuss that more in a few slides time.



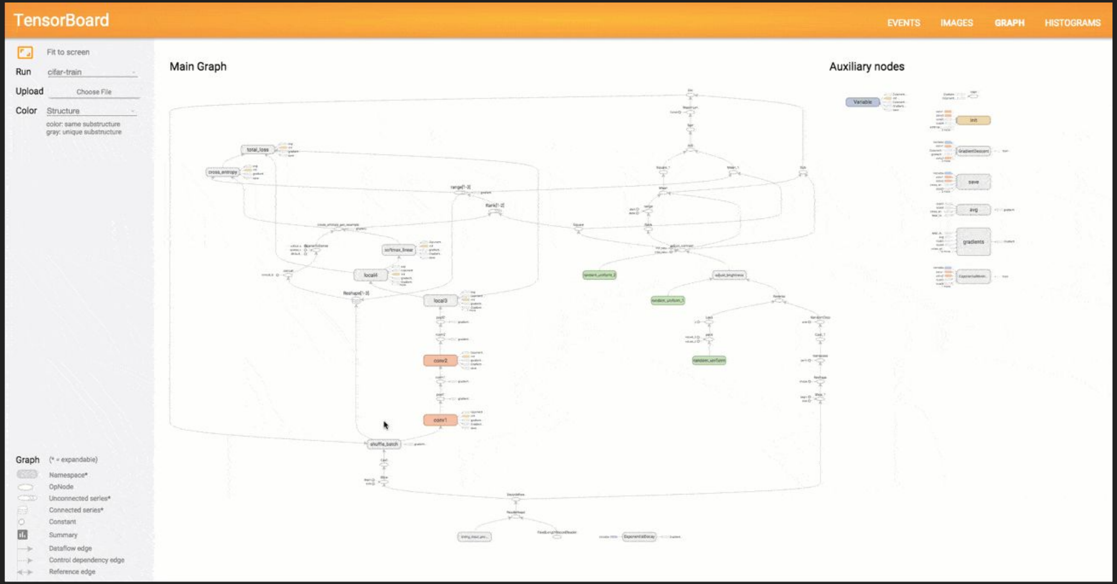
```
In [1]: import tensorflow as tf
        import numpy as np
```

```
In [2]: x = tf.Variable(np.ones(shape=(10,)))
        y = tf.Variable(np.ones(shape=(10,)))
        z = x + y
        y_plus_z = tf.assign_add(y, z)
        init = tf.global_variables_initializer()
        with tf.Session() as sess:
            writer = tf.summary.FileWriter("/tmp/graph", sess.graph)
            sess.run(init)
            print(sess.run(y_plus_z))
            print(sess.run(y_plus_z))
            print(sess.run(y_plus_z))
            writer.close()
```

```
[ 3.  3.  3.  3.  3.  3.  3.  3.  3.  3.]
[ 7.  7.  7.  7.  7.  7.  7.  7.  7.  7.]
[ 15.  15.  15.  15.  15.  15.  15.  15.  15.  15.]
```



Larger example



Source: https://www.tensorflow.org/get_started/graph_viz

What is Dryad?

“Dryad is a general-purpose distributed execution engine for coarse-grain data-parallel applications. A Dryad application combines computational “vertices” with communication “channels” to form a dataflow graph. Dryad runs the application by executing the vertices of this graph on a set of available computers, communicating as appropriate through files, TCP pipes, and shared-memory FIFOs. The vertices provided by the application developer are quite simple and are usually written as sequential programs with no thread creation or locking. Concurrency arises from Dryad scheduling vertices to run simultaneously on multiple computers, or on multiple CPU cores within a computer”

Isard, M., Budiu, M., Yu, Y., Birrell, A., & Fetterly, D. (2007, March). Dryad: distributed data-parallel programs from sequential building blocks. In ACM SIGOPS operating systems review (Vol. 41, No. 3, pp. 59-72). ACM.

What is Tensorflow?

Theano with inter-device communication as a first class citizen.

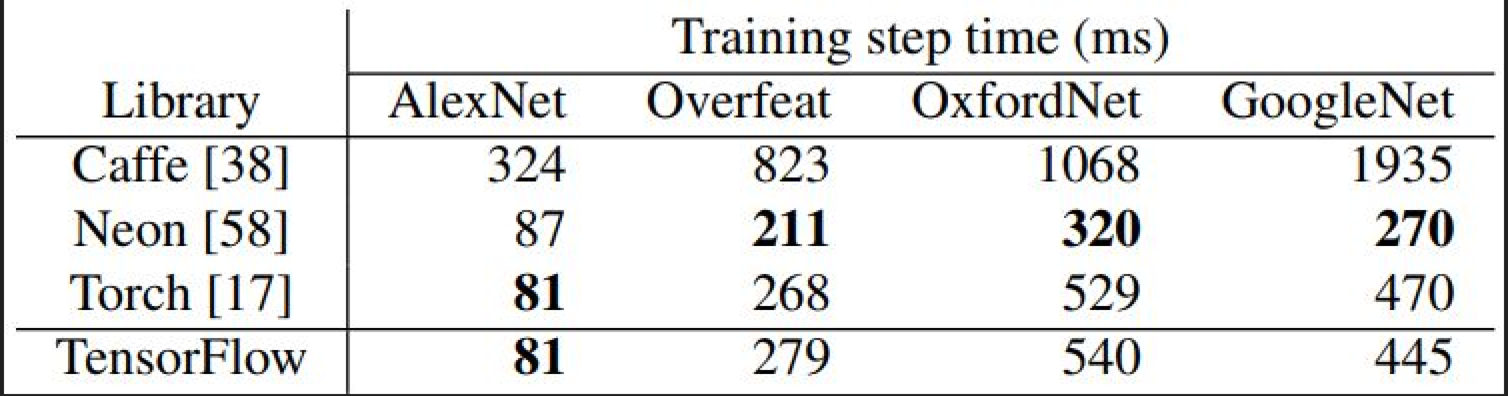
Send and Recv operations (nodes in the graph) with specific implementations for particular device pairs.

GPU-GPU? Use DMA.

Host-Host? Networked implementation.

Results

Competitive on a single machine:

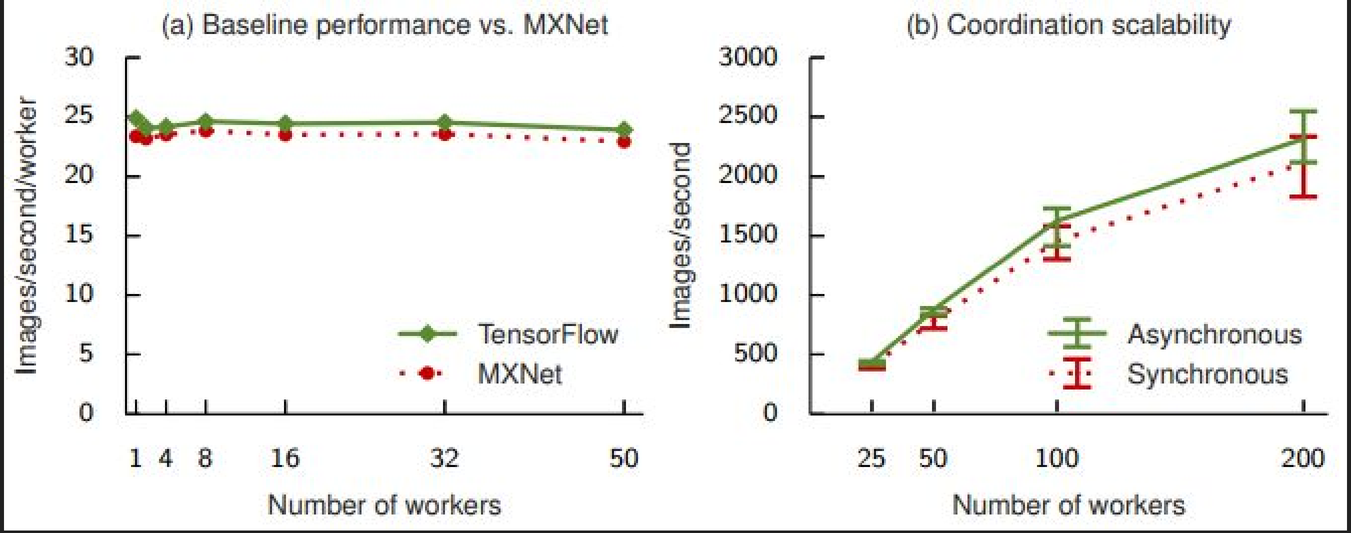


	Training step time (ms)			
Library	AlexNet	Overfeat	OxfordNet	GoogleNet
Caffe [38]	324	823	1068	1935
Neon [58]	87	**211**	**320**	**270**
Torch [17]	**81**	268	529	470
TensorFlow	**81**	279	540	445

Source: Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016, November). TensorFlow: A System for Large-Scale Machine Learning. In OSDI (Vol. 16, pp. 265-283).
Chicago

Results

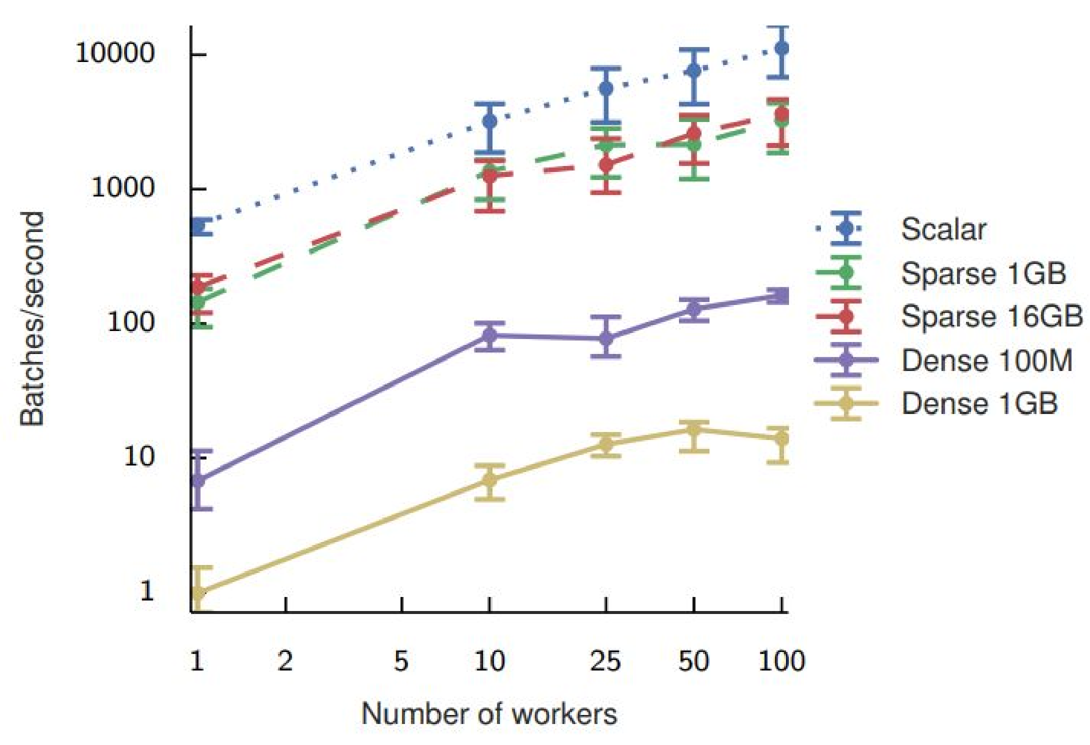
Deployable on a cluster:



Source: Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016, November). TensorFlow: A System for Large-Scale Machine Learning. In OSDI (Vol. 16, pp. 265-283).
Chicago

Results

Note that sparse updates of this style were initially developed in Project Adam.



Source: Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016, November). TensorFlow: A System for Large-Scale Machine Learning. In OSDI (Vol. 16, pp. 265-283).
Chicago

My thoughts

Relatively little theoretical or ideological novelty - but *___extremely pragmatic, well executed and useful___*.

They understood the problem domain well, specifically the relaxed consistency constraints that allow for faster weight propagation than Spark and the power of a Theano-style API.

Theano is dead [0], long live Tensorflow.

One criticism - is the Tensor itself limiting? Users must work around the lack of ragged dimensions.

[0] Announcement of the end of development. https://groups.google.com/forum/#!msg/theano-users/7Poq8BZutbY/rNCIfvAEAwAJ