TENSORFLOW: A SYSTEM FOR LARGE-SCALE MACHINE LEARNING

AUTHORS: MARTÍN ABADI, PAUL BARHAM, JIANMIN CHEN, ZHIFENG CHEN, ANDY DAVIS, JEFFREY DEAN, MATTHIEU DEVIN, SANJAY GHEMAWAT, GEOFFREY IRVING, MICHAEL ISARD, MANJUNATH KUDLUR, JOSH LEVENBERG, RAJAT MONGA, SHERRY MOORE, DEREK G. MURRAY, BENOIT STEINER, PAUL TUCKER, VIJAY VASUDEVAN, PETE WARDEN, MARTIN WICKE, YUAN YU, AND XIAOQIANG ZHENG
OVERVIEW

• Large Scale ML System
• Distributed Compute and Training
  • Multi-node
  • Heterogenous Environments
• Dataflow Graphs
• Open Source
• Mathematically Flexible
  • Bespoke Loss & Kernels
• Fault Tolerant
DATAFLOW GRAPHS

Input 1 ➔ Multiply ➔ Add ➔ Input 3 ➔ Output

Mutability!
PRIOR WORK

• DistBelief
  • Architecture
    • Parameter Server
    • Workers
  • Inflexible Layers
  • Inflexible Training Algorithms
    • RNNs, LSTMs, GCNs challenging
  • Optimized for large clusters

• Caffe & Theano
  • Similar

TensorFlow is designed to improve flexibility!
ACCELERATOR ABSTRACTION

CPU

GPU

TPU
UNITS OF TENSORFLOW

• Graph
  • Subgraph
  • Edges
  • Tensors
  • Vertices
  • Operations

Partitioned subgraphs are distributed to individual compute devices
Multidimensional arrays
Add, Multiply, Sigmoid

• Automatic Partitioning
  • Subgraphs distributions maximize compute efficiency
CONTROL FLOW

- Graph Partitioned and Distributed
- Send + Recv Replace Split Edges
- Send
  - Pushes value from one device to another
- Recv
  - Blocks until value available
- “Deferred execution”

EXECUTION

- Synchronous Execution
  - Classically frowned upon
  - GPUs make appealing
- All workers forced to take same parameters
- Backup workers stochastically eliminate straggling processes
DIFFERENTIATION & BACKPROP

- Symbolic representation
  - Automatically computes backprop code
- Like PS architectures, enables distributed training via +/- write operations
IMPLEMENTATION

Figure 6: The layered TensorFlow architecture.
### SINGLE MACHINE BENCHMARKS

<table>
<thead>
<tr>
<th>Library</th>
<th>AlexNet</th>
<th>Overfeat</th>
<th>OxfordNet</th>
<th>GoogleNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caffe [38]</td>
<td>324</td>
<td>823</td>
<td>1068</td>
<td>1935</td>
</tr>
<tr>
<td>Neon [58]</td>
<td>87</td>
<td>211</td>
<td>320</td>
<td>270</td>
</tr>
<tr>
<td>Torch [17]</td>
<td>81</td>
<td>268</td>
<td>529</td>
<td>470</td>
</tr>
<tr>
<td>TensorFlow</td>
<td>81</td>
<td>279</td>
<td>540</td>
<td>445</td>
</tr>
</tbody>
</table>
Figure 7: Baseline throughput for synchronous replication with a null model. Sparse accesses enable TensorFlow to handle larger models, such as embedding matrices (§4.2).
CNN IMPLEMENTATIONS

<table>
<thead>
<tr>
<th>Library</th>
<th>AlexNet</th>
<th>Overfeat</th>
<th>OxfordNet</th>
<th>GoogleNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caffe [38]</td>
<td>324</td>
<td>823</td>
<td>1068</td>
<td>1935</td>
</tr>
<tr>
<td>Neon [58]</td>
<td>87</td>
<td>211</td>
<td>320</td>
<td>270</td>
</tr>
<tr>
<td>Torch [17]</td>
<td>81</td>
<td>268</td>
<td>529</td>
<td>470</td>
</tr>
<tr>
<td>TensorFlow</td>
<td>81</td>
<td>279</td>
<td>540</td>
<td>445</td>
</tr>
</tbody>
</table>

Table 1: Step times for training four convolutional models with different libraries, using one GPU. All results are for training with 32-bit floats. The fastest time for each model is shown in bold.
SYNC AND NON-SYNCHED PROCESSES

Figure 8: Results of the performance evaluation for Inception-v3 training (§6.3). (a) TensorFlow achieves slightly better throughput than MXNet for asynchronous training. (b) Asynchronous and synchronous training throughput increases with up to 200 workers. (c) Adding backup workers to a 50-worker training job can reduce the overall step time, and improve performance even when normalized for resource consumption.
Figure 9: Increasing the number of PS tasks leads to increased throughput for language model training, by parallelizing the softmax computation. Sampled softmax increases throughput by performing less computation.
CRITICISM

• No actual accuracy comparisons
• Convergence comparisons in synchrony analysis?
• Lacking capability for abstracted computation
  • Reason why Keras runs on top of TF
CONCLUSION

• Built a ML system that is:
  • Robust
  • Distributable
  • Extensible
  • Fast

• In the ensuing years
  • Used extensively
  • Extended
• TensorFlow: A System for Large-Scale Machine Learning