

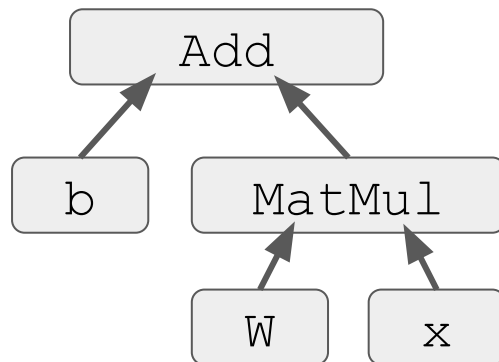
TensorFlow Debugger: Debugging Dataflow Graphs for Machine Learning

Shanqing Cai, Eric Breck, Eric Nielsen, Mike Salib, D. Sculley
{cais, ebreck, nielsene, msalib, dsculley}@google.com



TensorFlow Debugger (tfdbg): Why do we need it?

- Software systems all need debugging
 - ML systems are no exceptions.
- ML systems are numerical and data-driven:
 - Bugs are more obscure and subtle.
- TensorFlow uses dataflow graph paradigm
 - Debugging is even more difficult.



TensorFlow graph execution: A black box

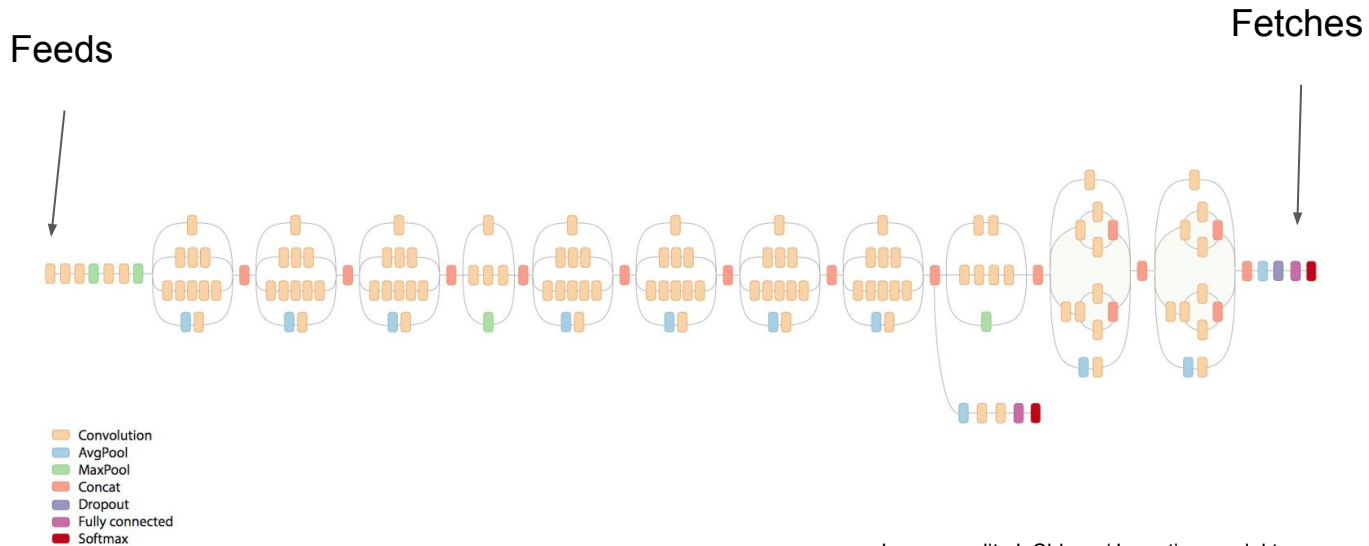
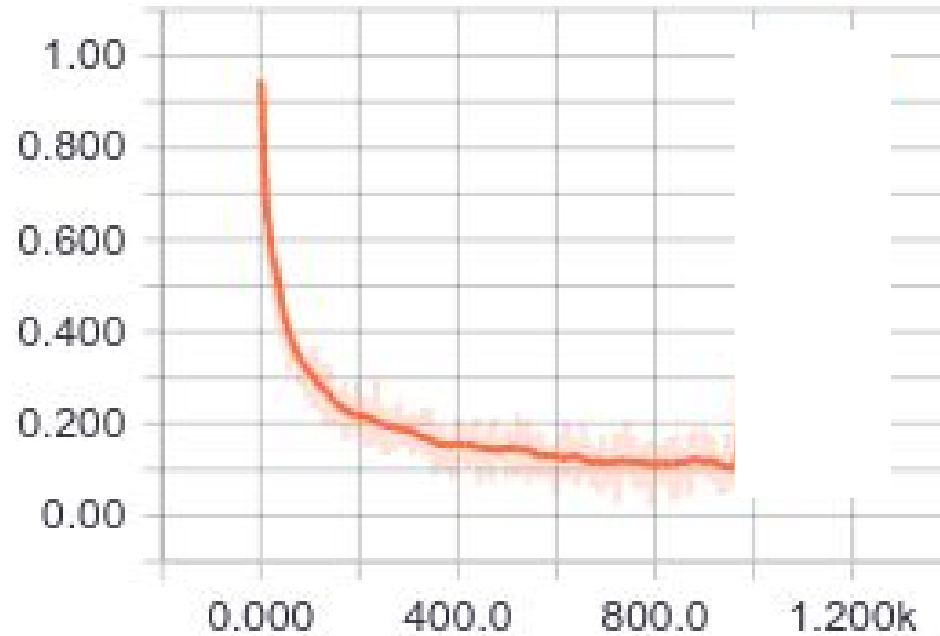


Image credit: J. Shlens / Inception model team
<https://research.googleblog.com/2016/03/train-your-own-image-classifier-with.html>

- Only the inputs (feeds) and output (fetches) are visible.
- The internal states are abstracted away in a function call.

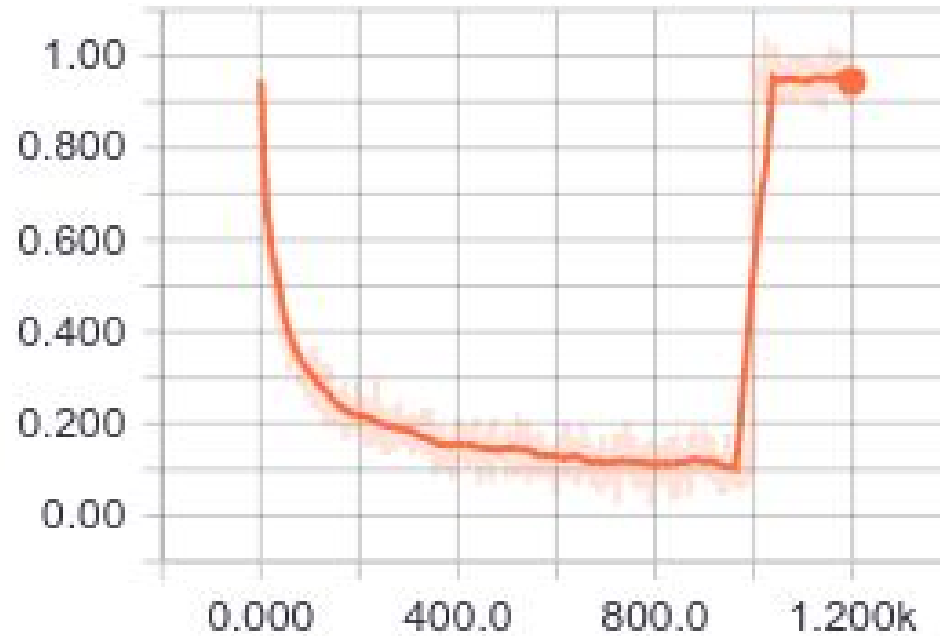
TensorFlow graph execution: A black box

Validation error



TensorFlow graph execution: A black box

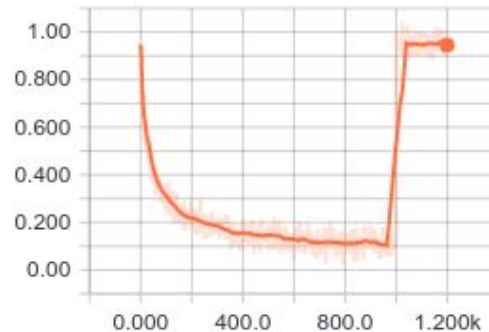
Validation error



TensorFlow graph execution: A black box

Example bug: Training suddenly fails

Validation error




- When issues occur, it is generally difficult to know which node of the graph is responsible, especially for large models.
- Conventional debuggers are not super useful:
 - Python *pdb* does not provide visibility to the level of C++ ops.
 - *gdb* is not capable of organizing the data in a form relevant to graph elements and states.


Examples of common issues to debug

- Bad numerical values (infinities / NaNs) due to underflow/overflow, division by zero, log of zero, etc.
- Tensor shape mismatches
- Incorrect preprocessing of input data, normalization, hashing, ...
- Concurrency issues on the graph
- Bugs in ops
- Bugs introduced by changes in TensorFlow
- Add your own debugging story ...

How do you debug the black box

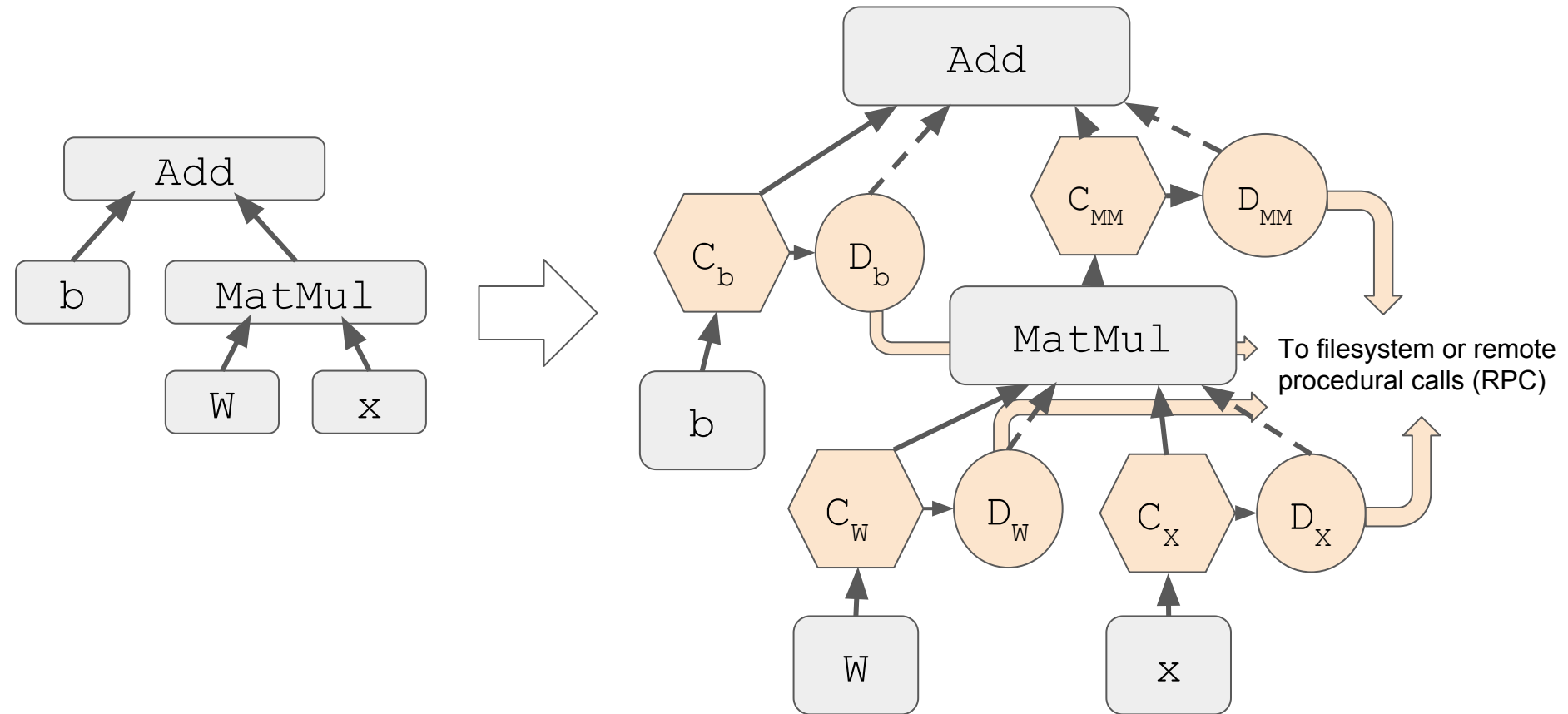
Option	Amount of code change	Coverage of nodes	User interfaces
Manual fetch and print	Large	One at a time	Non-interactive console
Print ops	Small	One at a time	Non-interactive console
Summary ops	Medium	One at a time	Browser (TensorBoard)
 tfdbg	Small	Full graph or any chosen subgraph	Iterative CLI + TensorBoard (developing)

How do you debug the black box

Option	Amount of code change	Coverage of nodes	User interfaces
Manual fetch and print	Large	One at a time	Non-interactive console
Print ops	Small	One at a time	Non-interactive console
Summary ops	Medium	One at a time	Browser (TensorBoard)
 tfdbg	Small	Full graph or any chosen subgraph	Iterative CLI + TensorBoard (developing)

```
from tensorflow.python import debug as tfdbg  
  
sess = tfdbg.LocalCLIDebugWrapperSession(sess)
```

To export intermediate tensors, the **analyzer** inserts copy (C) & debug (D) nodes to the graph.



- The RPC option supports debugging of remotely running sessions (e.g., in distributed training).

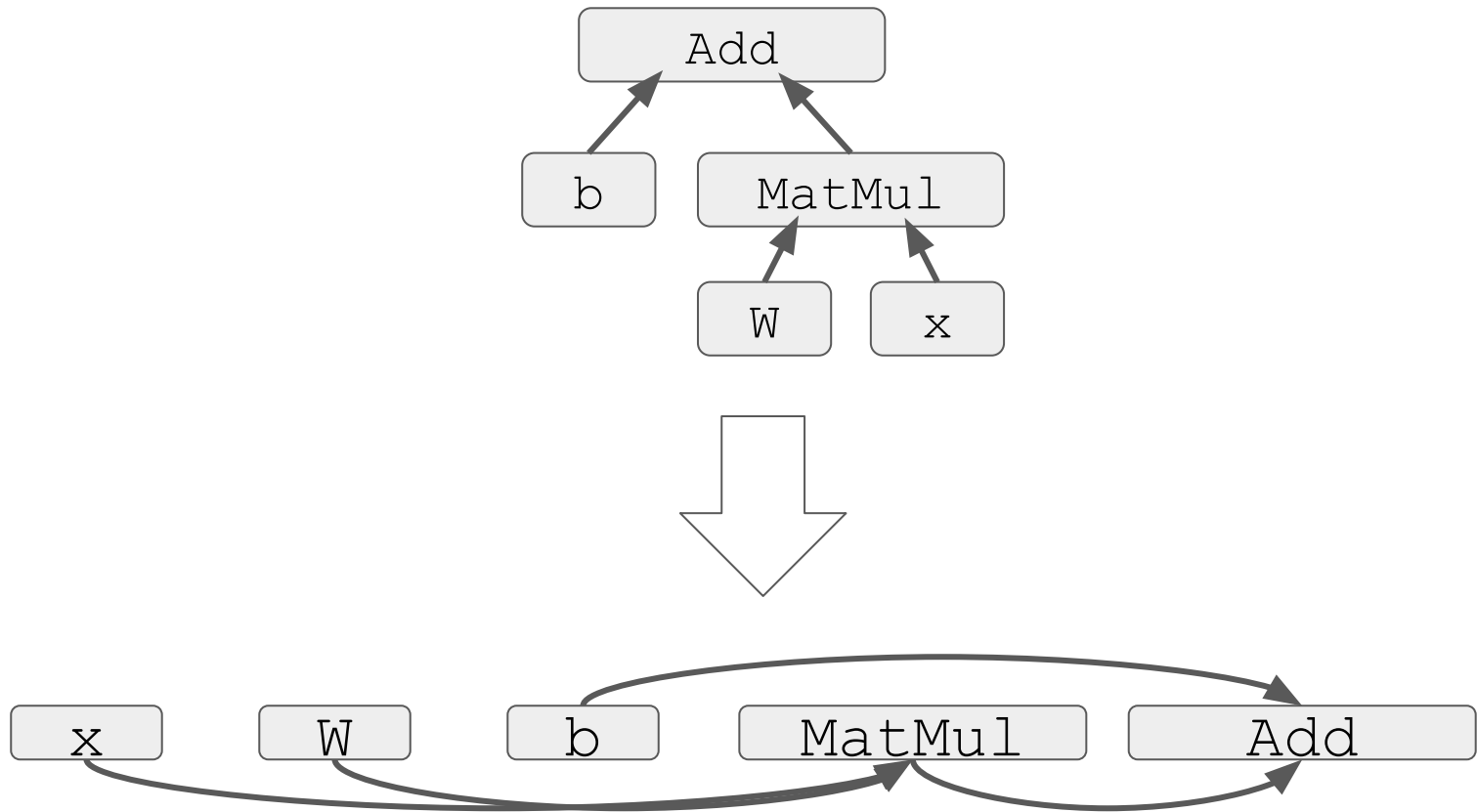
tfdbg analyzer

Command-Line Interface:

- Exposes graph structure
- Exposes node attributes
- Exposes intermediate tensors
- *Conditional breakpoints* across runs
(Run till certain issues happen)

```
run-end: run #4: 1 fetch (train/Adan); 2 feeds -----
[15,4] -2.98563995e+01, -2.45755520e+01, -2.41641159e+01, -1.01,
-5.49377441e+01, -4.12961807e+01],
[ -5.88836009e-03, -6.49229479e+00, -9.37573051e+00, -3.20481339e+01,
-9.56073856e+00, -5.93430567e+00, -6.46704340e+00, -6.74948425e+01,
-3.04495449e+01, -2.66743145e+01],
[ -1.16668917e-01, -1.02749815e+01, -1.22198334e+01, -3.23261566e+01,
-2.87913720e+00, -5.80756617e+00, -2.97803330e+00, -5.92251053e+01,
-3.10324211e+01, -2.21582317e+01],
[ -1.24847364e+00, -1.26194906e+00, -7.56814337e+00, -2.49623947e+01,
-9.67004299e-01, -3.13249445e+00, -5.18321943e+00, -6.07339478e+01,
-3.12824078e+01, -1.88571663e+01],
[ -4.26952261e-03, -8.45148849e+00, -1.34998159e+01, -2.91649876e+01,
-6.77460957e+00, -5.85078144e+00, -1.05798416e+01, -6.64147110e+01,
-2.70646992e+01, -2.65593338e+01],
[ 0.00000000e+00, -2.84139137e+01, -2.81213684e+01, -4.82183952e+01,
-2.76676750e+01, -2.23679295e+01, -2.53318958e+01, -1.01,
-5.41935539e+01, -4.11526871e+01],
[ -6.56682968e+00, -9.56997490e+00, -1.60626431e+01, -3.57540321e+01,
-1.27452021e+01, -9.74107265e+00, -1.53910276e-03, -6.40680313e+01,
-3.60007935e+01, -3.14809284e+01],
[21,8]
--- Scroll (PgDn/PgUp): 16.50% [15,4]-[21,8] -----
tfdbg>
```

To allow stepping, the **node-stepper** topologically sorts the nodes.



tfdbg node-stepper

Command-Line Interface:

- Allows stepping or continuing through the nodes in a linear fashion
- Reuses previous results to minimize re-computation
- Allows injecting (overriding) values in the middle of the debugging

```
--- run-start: run #1: 1 fetch (accuracy/accuracy/Mean:0); 2 feeds -----  
About to enter Session run() call #1:  
  
Fetch(es):  
  accuracy/accuracy/Mean:0  
  
Feed dict(s):  
  input/x-input:0  
  input/y-input:0  
-----  
Select one of the following commands to proceed ---->  
  run:  
    Execute the run() call with debug tensor-watching  
  run -n:  
    Execute the run() call without debug tensor-watching  
  run -f <filter_name>:  
    Keep executing run() calls until a dumped tensor passes a given, registered  
    filter (conditional breakpoint mode).  
    Registered filter(s):  
----- Scroll (PgDn): 0.00% -----  
tfdbg>
```

⇒ Controllability of the graph execution process

More Features of tfdbg:

- Integrated with `tf.contrib.learn` (A high-level ML API built on TensorFlow)
- C++ and Python API for probing tensors in production
- White-box testing of models

Developing:

- Integration with TensorBoard for visual debugging.
- Better support for distributed sessions.

Summary

- TensorFlow Debugger (tfdbg) is a specialized debugger for the dataflow graph-based ML library.
- It provides both observability and controllability to the graph execution process.
- This leads to
 - A unified, powerful and easy-to-use debugging workflow
 - Ability to do white-box testing of model behavior.

Available in open-source TensorFlow:
tensorflow/python/debug

<https://github.com/tensorflow/tensorflow/tree/master/tensorflow/python/debug/examples>



Thank You!

Thanks also to the TensorFlow team and related teams for reviewing and guiding the design of TensorFlow Debugger.

