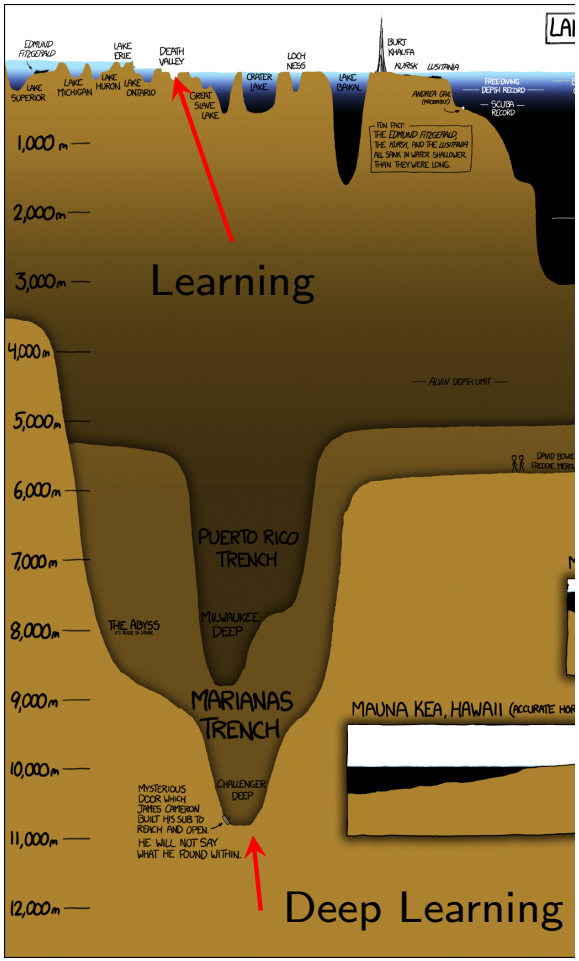
A Tour of TensorFlow

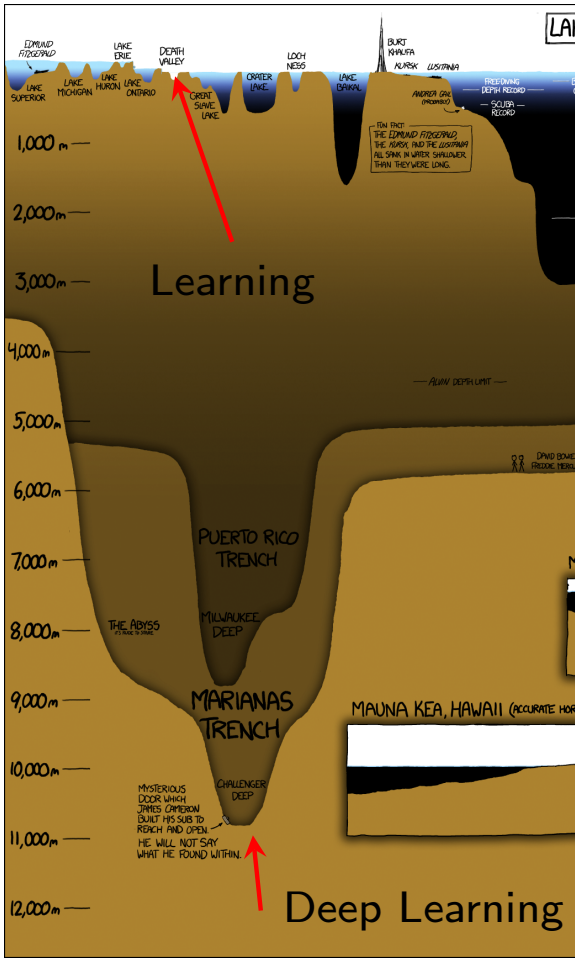
Peter Goldsborough

May 31, 2016

A Tour of TensorFlow

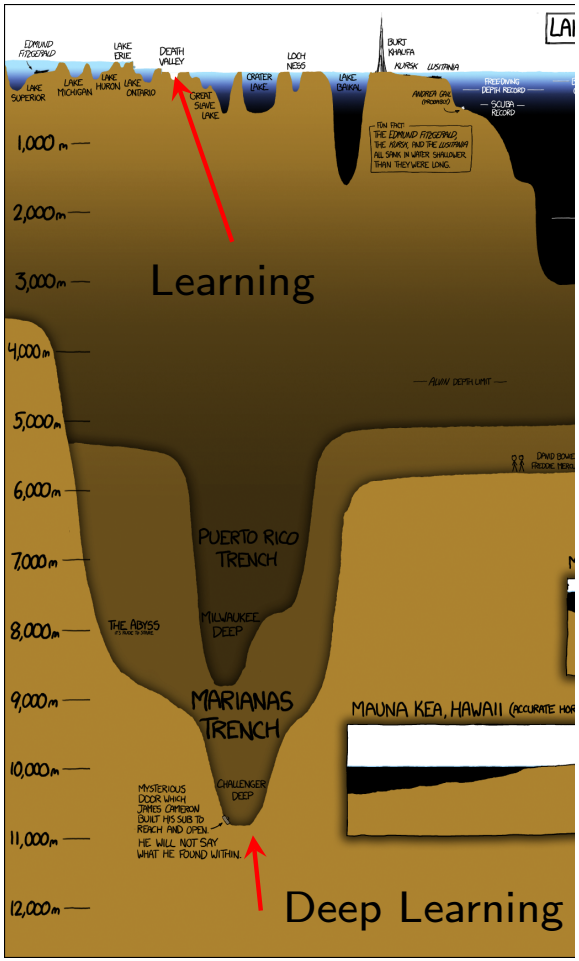


A Tour of TensorFlow



TensorFlow is

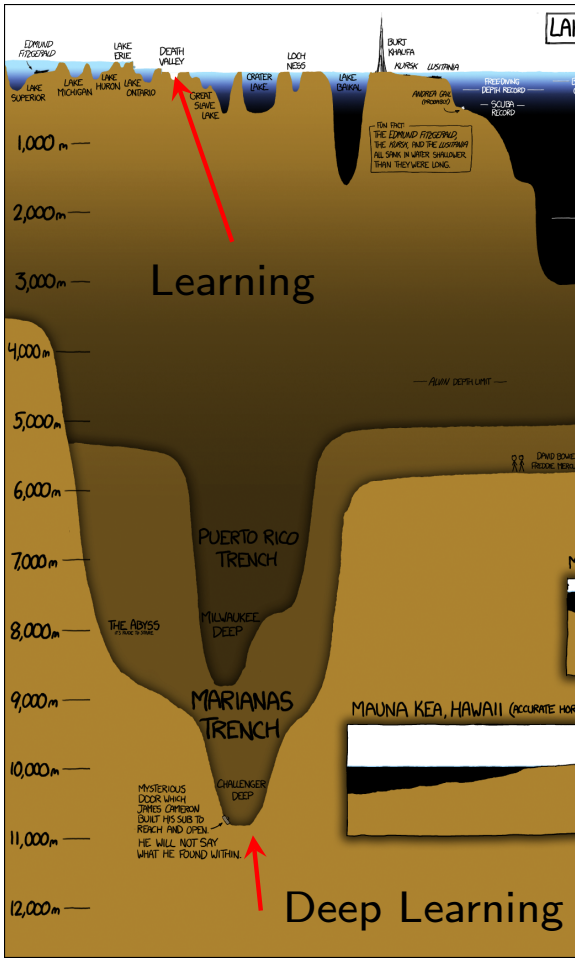
A Tour of TensorFlow



TensorFlow is

- An open source deep learning library

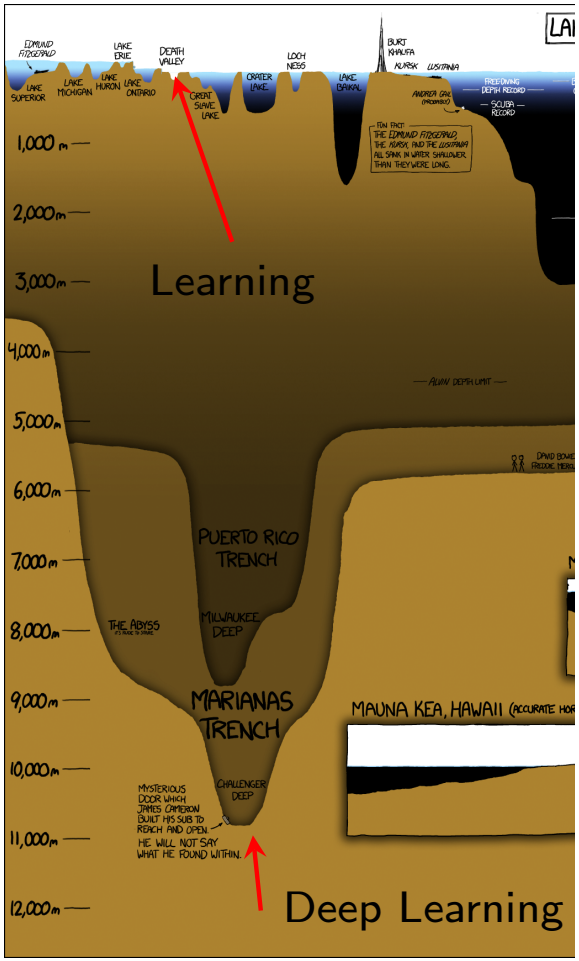
A Tour of TensorFlow



TensorFlow is

- An open source deep learning library
- Released by Google in November 2015

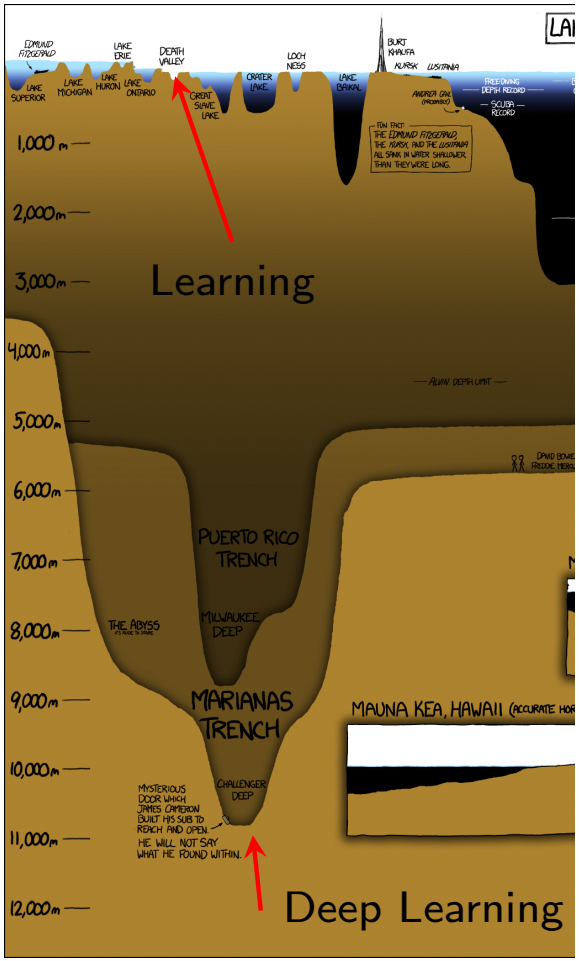
A Tour of TensorFlow



TensorFlow is

- An open source deep learning library
- Released by Google in November 2015
- Especially suited to:

A Tour of TensorFlow



TensorFlow is

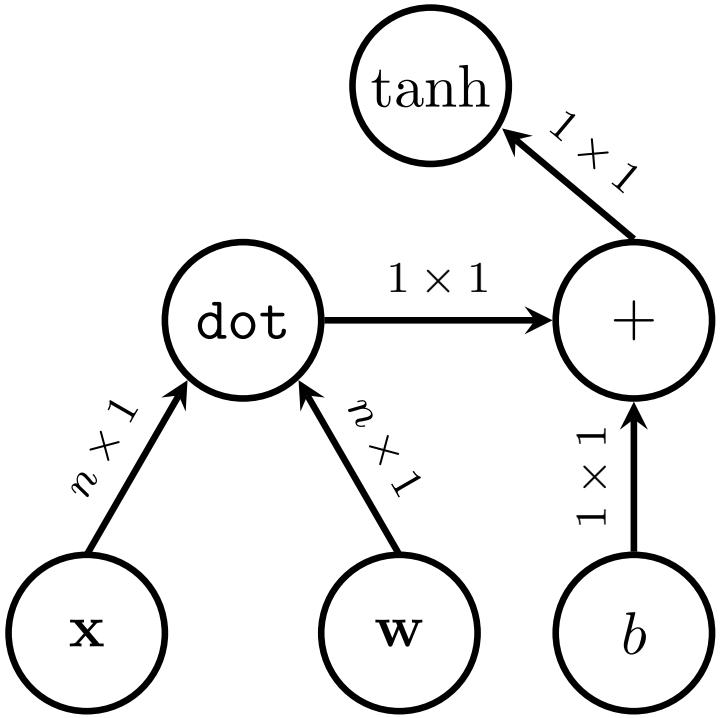
- An open source deep learning library
- Released by Google in November 2015
- Especially suited to:
  - "Large-scale machine learning on
  - heterogenous distributed systems"

Contents

1. Computational Paradigms
2. Execution Model
3. Back-Propagation in TensorFlow
4. Visualization Tools
5. Use Cases
6. Walkthrough

Computational Paradigms

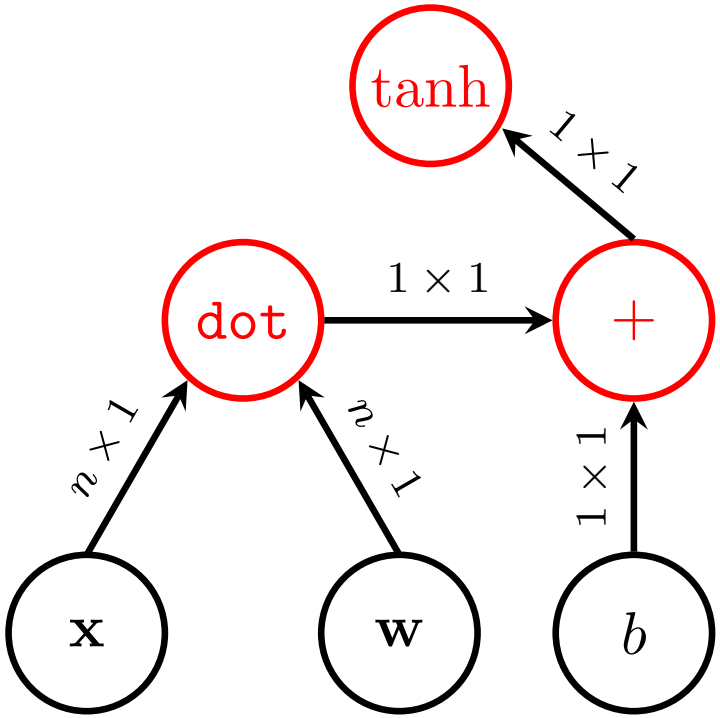
Computational Paradigms



$$\hat{y} = \tanh(\mathbf{x}^\top \mathbf{w} + b)$$

Computational Graphs

Computational Paradigms

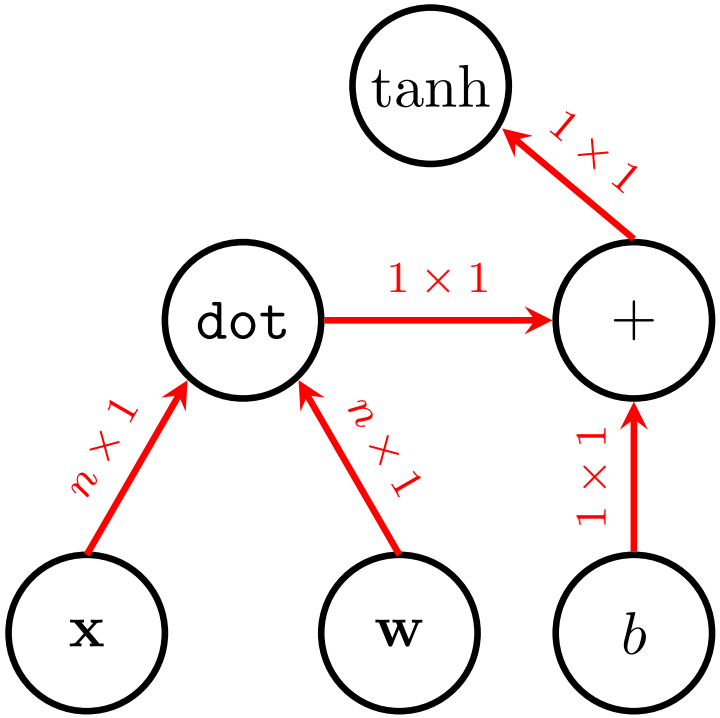


$$\hat{y} = \tanh(\mathbf{x}^\top \mathbf{w} + b)$$

Computational Graphs

1. Operations

Computational Paradigms

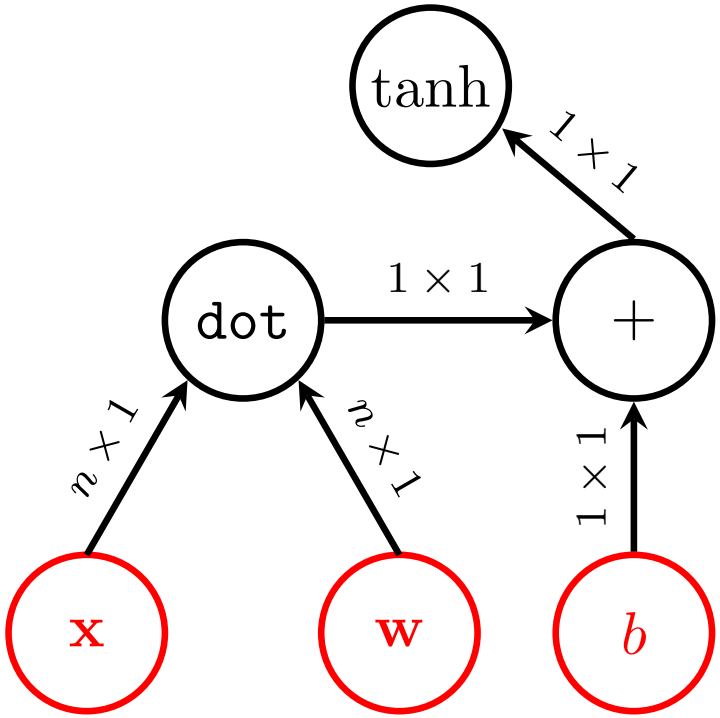


$$\hat{y} = \tanh(\mathbf{x}^\top \mathbf{w} + b)$$

Computational Graphs

1. Operations
2. Tensors

Computational Paradigms

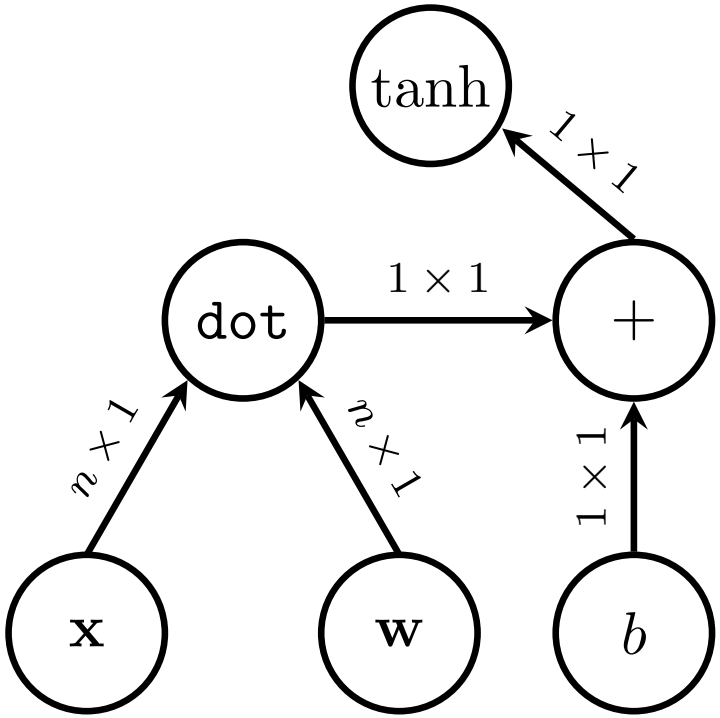


$$\hat{y} = \tanh(\mathbf{x}^\top \mathbf{w} + b)$$

Computational Graphs

1. Operations
2. Tensors
3. Variables

Computational Paradigms



$$\hat{y} = \tanh(\mathbf{x}^\top \mathbf{w} + b)$$

Computational Graphs

1. Operations
2. Tensors
3. Variables
4. Sessions

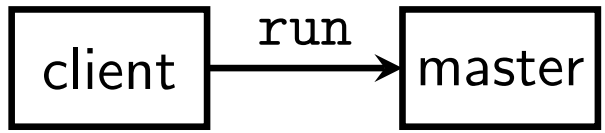
Execution Model

Execution Model

Actors

1. Client

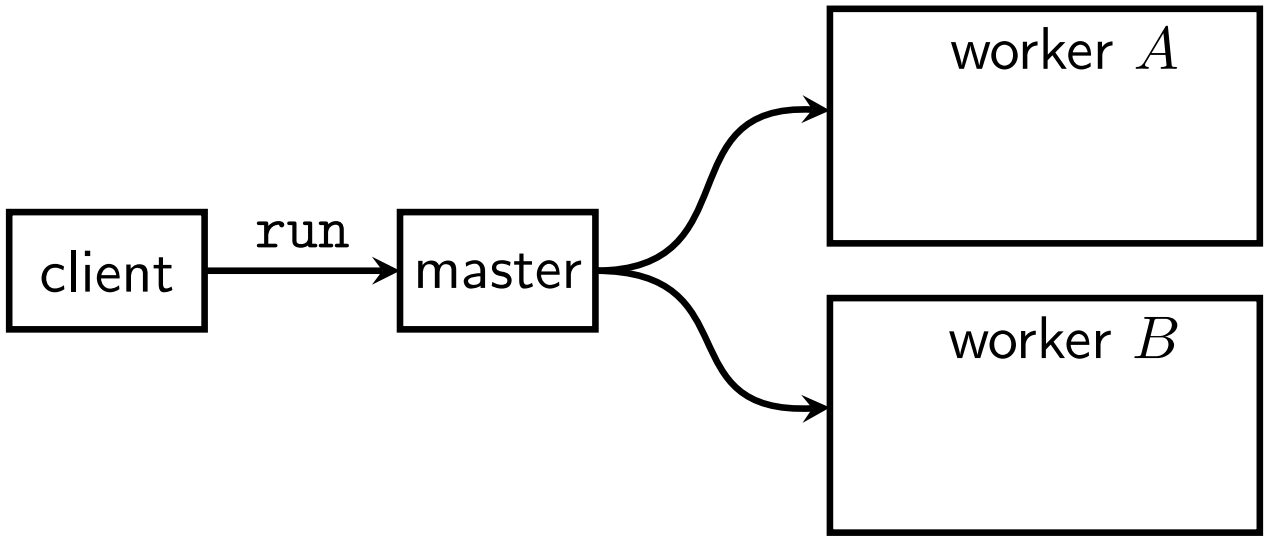
Execution Model



Actors

1. Client
2. Master

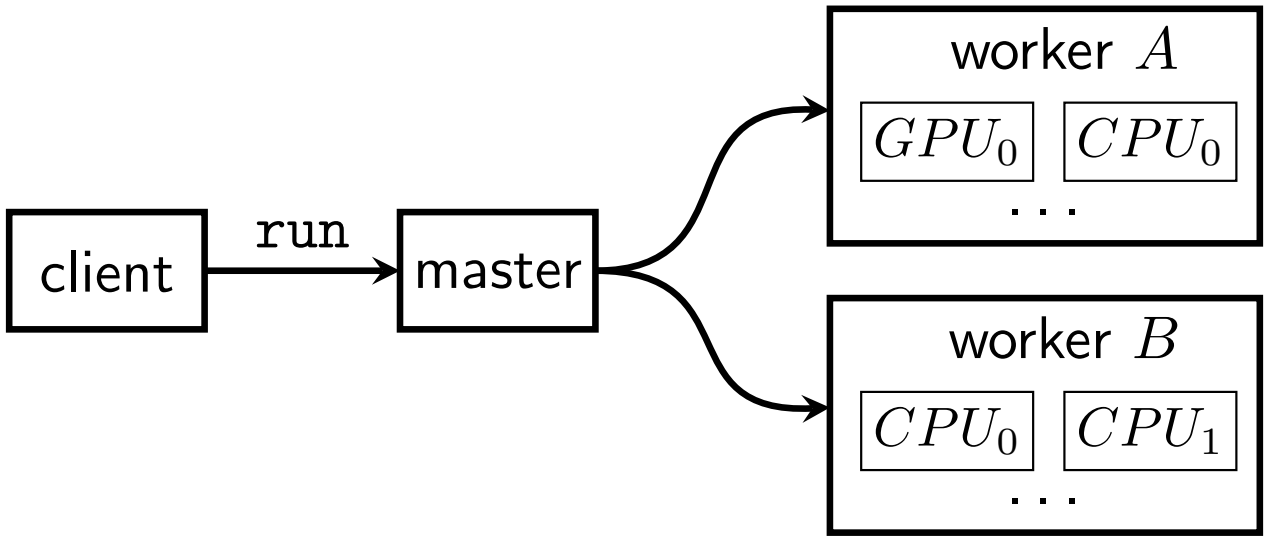
Execution Model



Actors

1. Client
2. Master
3. Workers

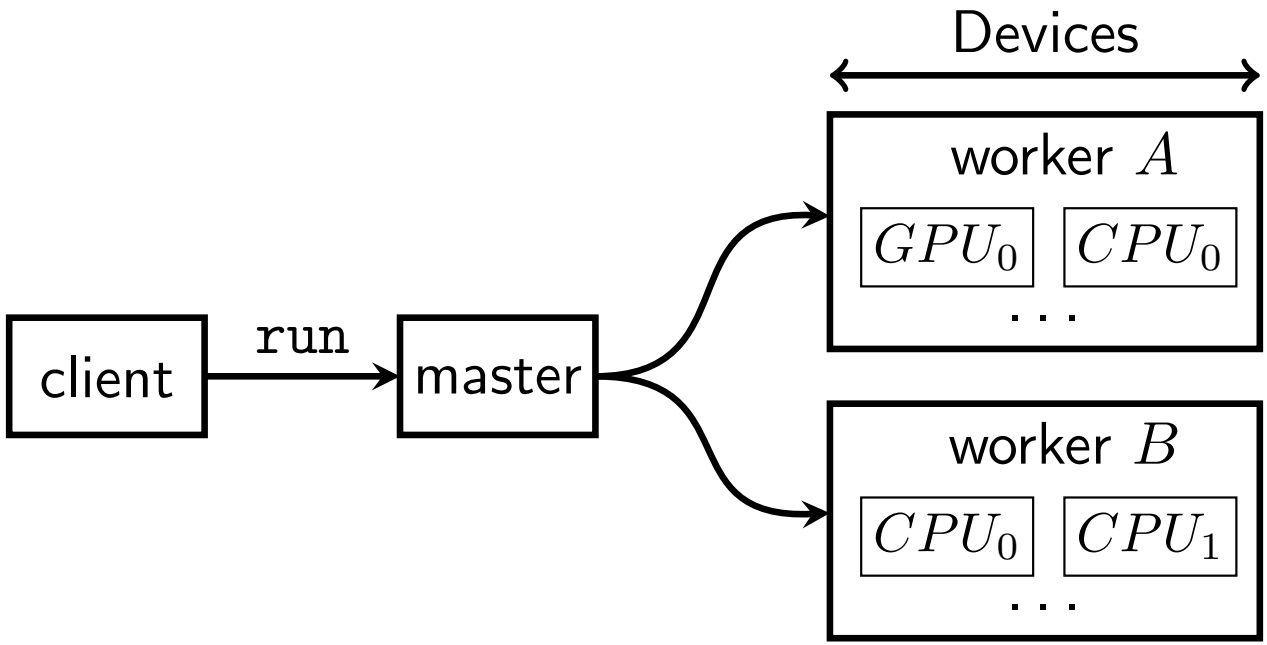
Execution Model



Actors

1. Client
2. Master
3. Workers
4. Devices

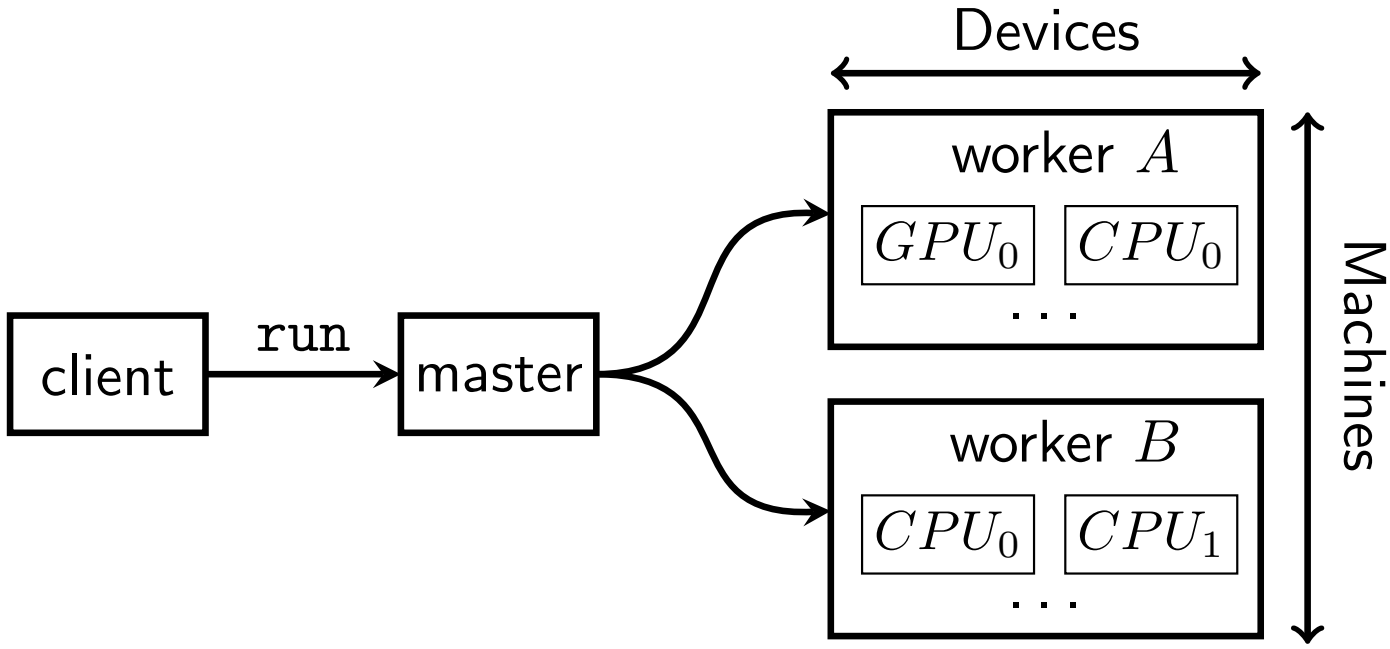
Execution Model



Actors

1. Client
2. Master
3. Workers
4. Devices

Execution Model

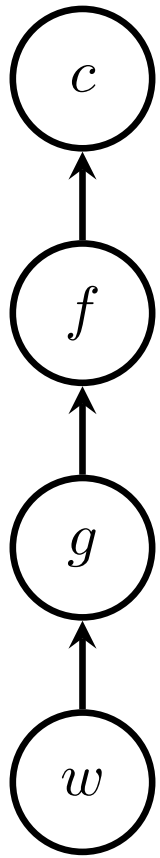


Actors

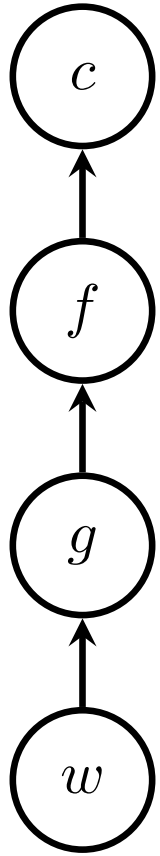
1. Client
2. Master
3. Workers
4. Devices

Back-Propagation in TensorFlow

Back-Propagation in TensorFlow



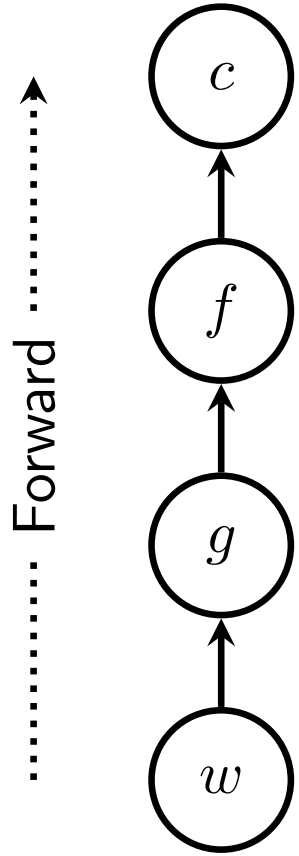
Back-Propagation in TensorFlow



Symbol to Number Differentiation

Back-Propagation in TensorFlow

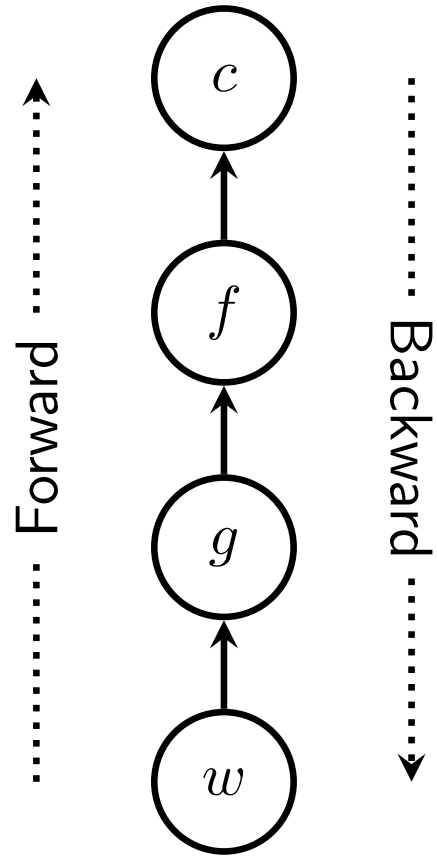
$$c = (f \circ g)(w)$$



Symbol to Number Differentiation

Back-Propagation in TensorFlow

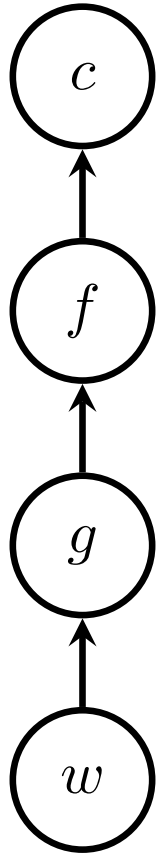
$$c = (f \circ g)(w)$$



$$\frac{\mathrm{d}c}{\mathrm{d}w} = f'(g(w)) \cdot g'(w)$$

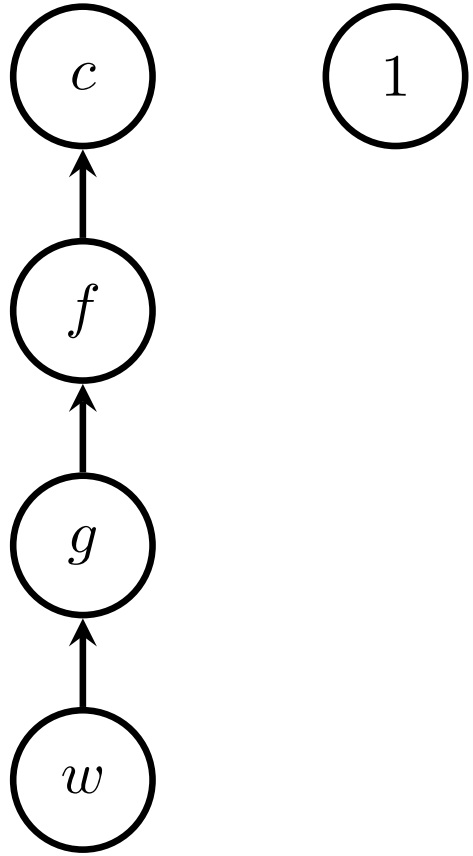
Symbol to Number Differentiation

Back-Propagation in TensorFlow



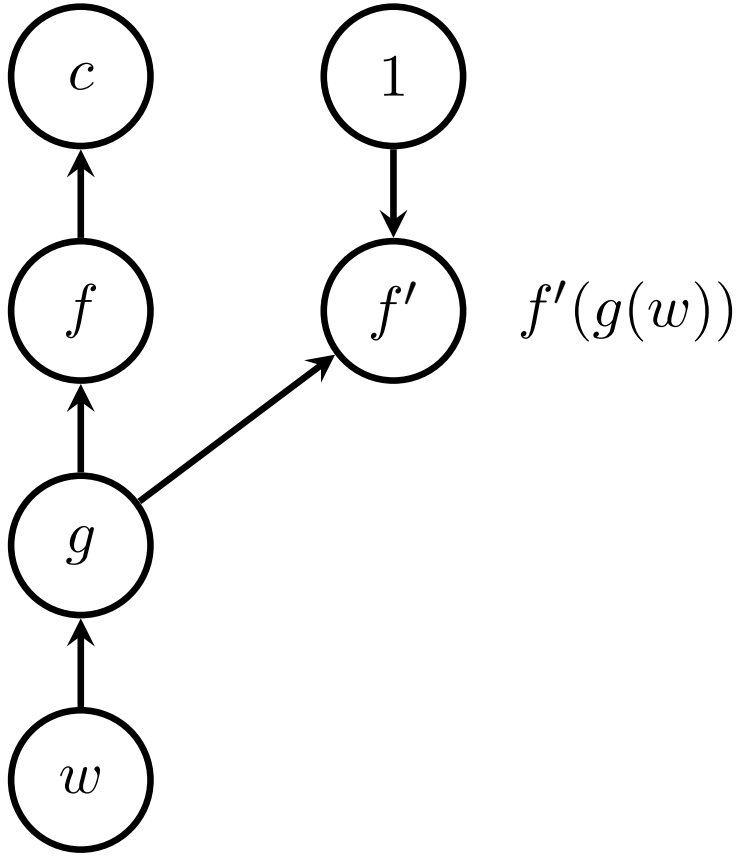
Symbol to Symbol Differentiation

Back-Propagation in TensorFlow



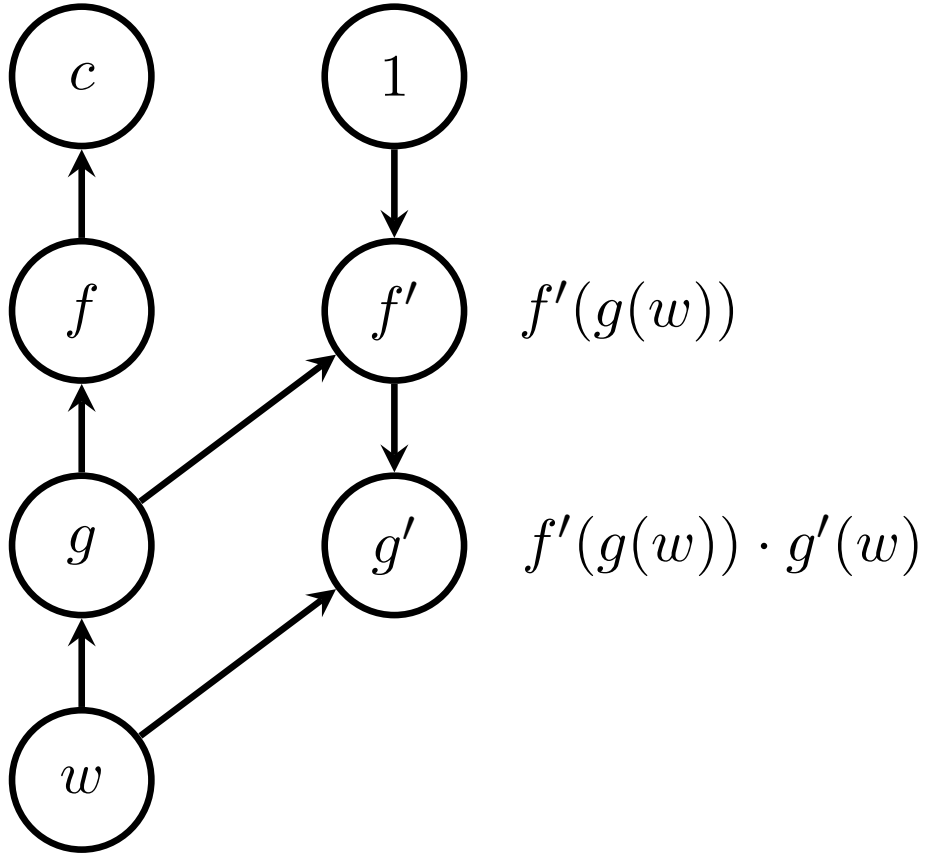
Symbol to Symbol Differentiation

Back-Propagation in TensorFlow



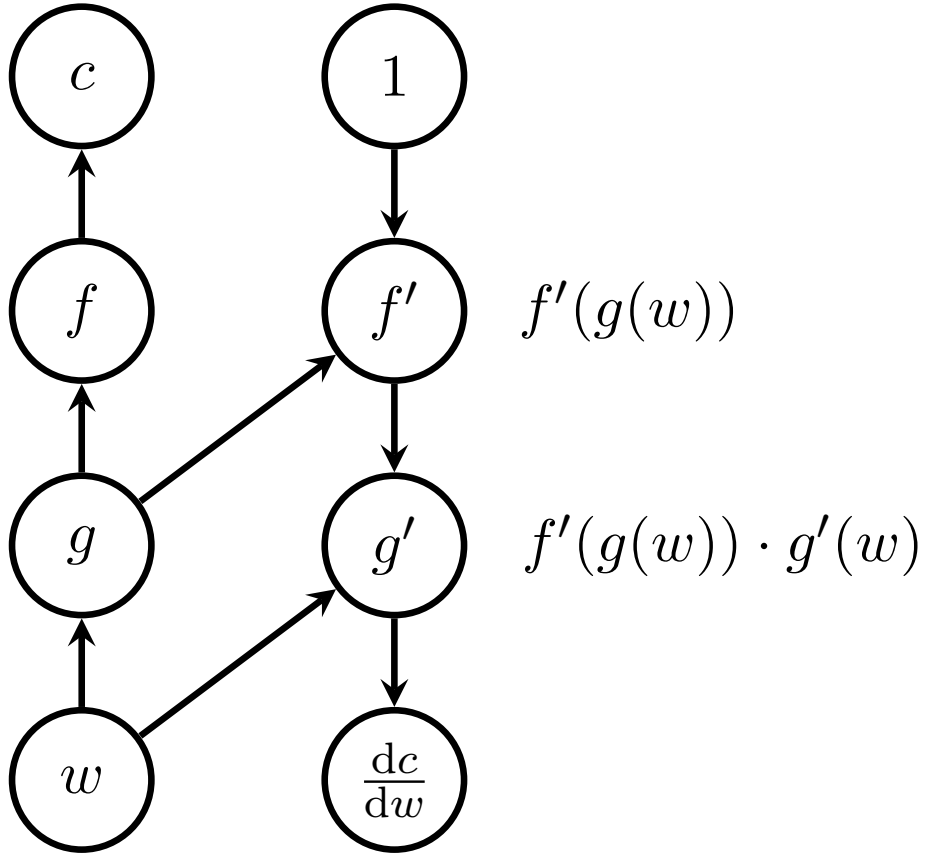
Symbol to Symbol Differentiation

Back-Propagation in TensorFlow



Symbol to Symbol Differentiation

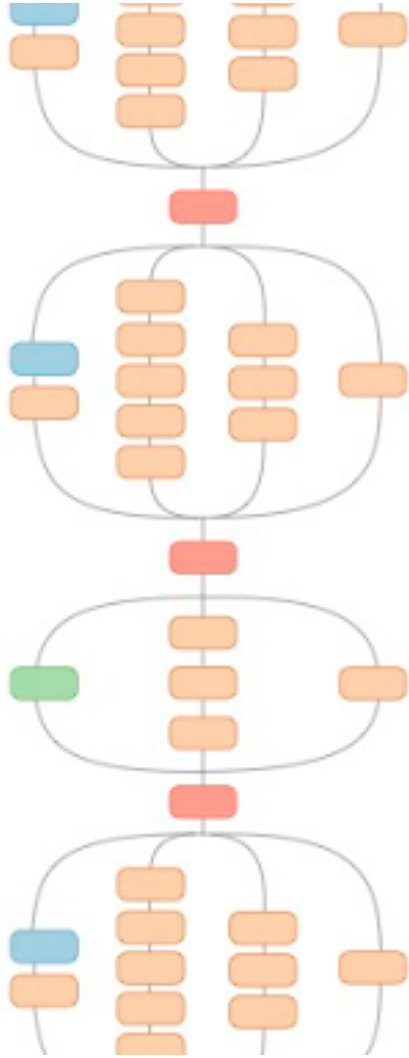
Back-Propagation in TensorFlow

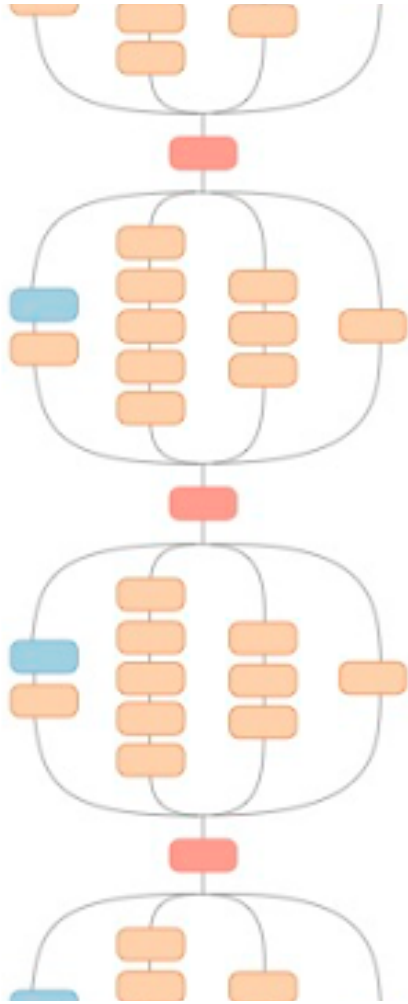


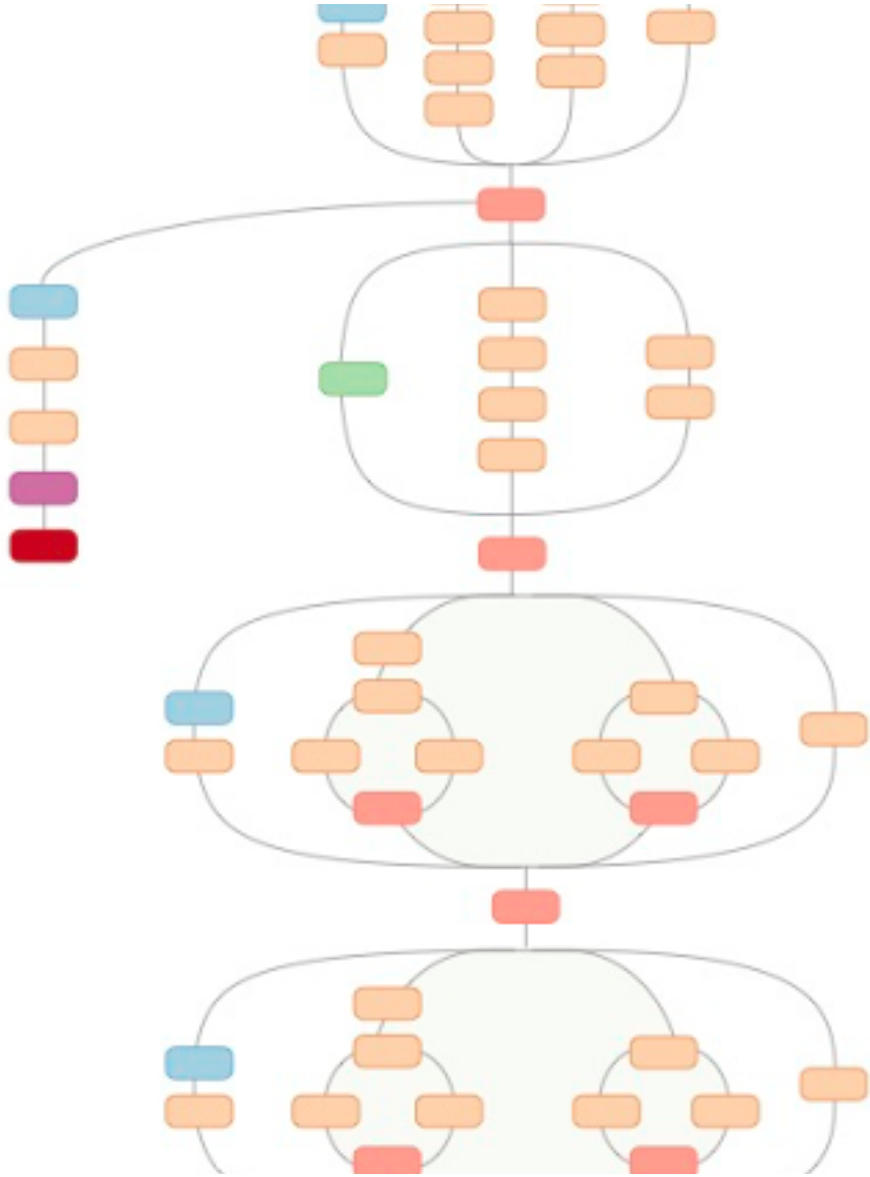
Symbol to Symbol Differentiation

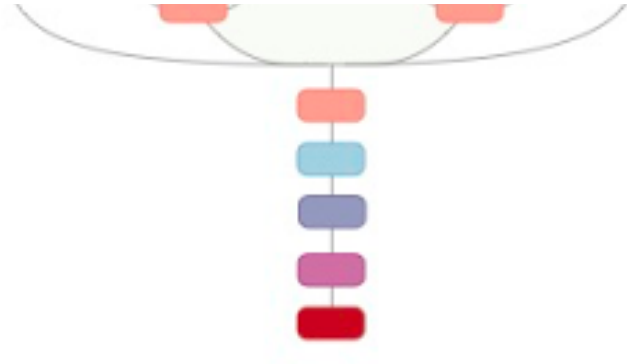
Visualization Tools

- Deep Neural Networks have the tendency of being . . . deep
- Easy to drown in the complexity of an architecture
- $>$ 36,000 nodes for Google's *Inception* model







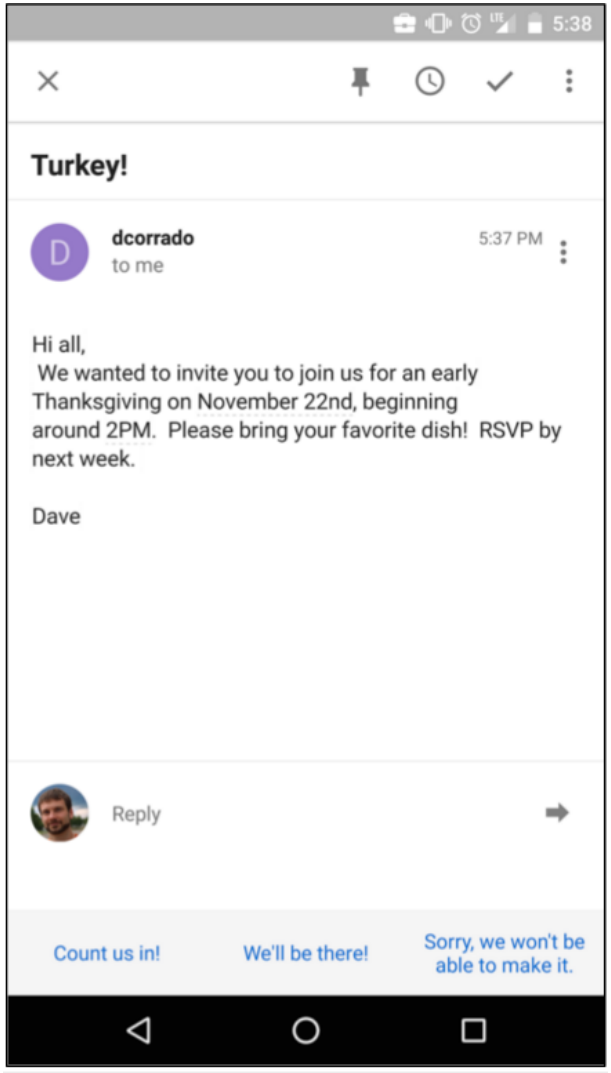


Source: http://googleresearch.blogspot.de/2016/03/train-your-own-image-classifier-with.html

TensorBoard to the Rescue

Use Cases

- Smart email replies in Google *Inbox*
- Emails mapped to "thought vectors"
- LSTMs synthesize valid replies



Source: http://googleresearch.blogspot.de/2015/11/computer-respond-to-this-email.html

Use Cases of TensorFlow

- Google DeepMind now using TensorFlow
- Already for *AlphaGo*
- According to a DeepMind SWE reasons are:
  - Integration with Google Cloud Platform,
  - Python,
  - Support for TPUs,
  - Ability to run on many GPUs.

Source: https://deepmind.com/css/images/opengraph/alphago-logo.png

Walkthrough

Thank You