TensorFlow

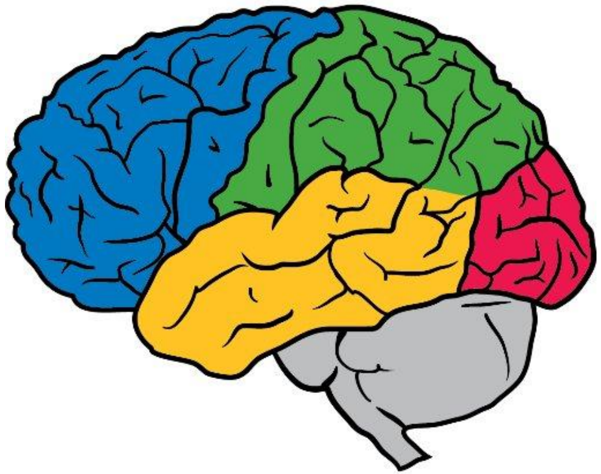
# Introduction to TensorFlow

Zholus Artem
ITMO University

Kaggle Club
2k17

# What is TensorFlow?

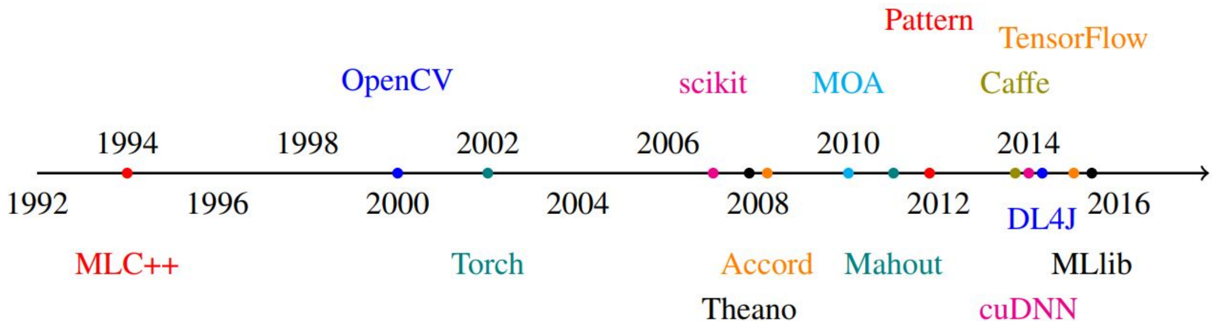
- Open Source Library
- Solves Machine Learning Problems (Not only Deep Learning)
- Distributed computing support
- Graph computation model
- Flexible Architecture



Kaggle Club
2017

# History and Motivation

- Scalable System
- Out-of-the-box CUDA and cuDNN support
- Easy to develop and deploy
- Debugger!



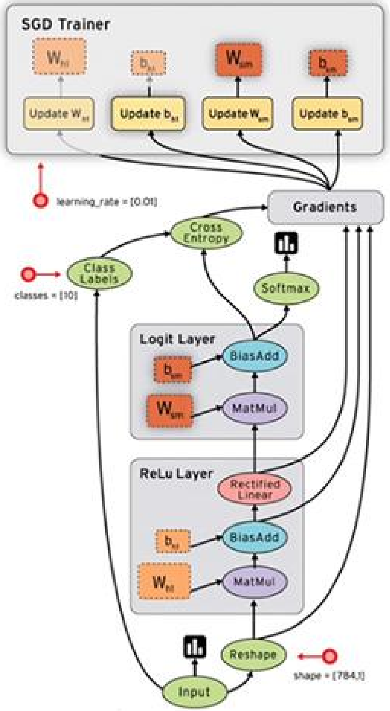
Kaggle Club
2017

* - picture from: A Tour of TensorFlow https://arxiv.org/abs/1610.01178

# Make Tensors Flow again!

## TF computational model

- Computational Graph (CG) is the way to express mathematical expressions symbolically
- Node in graph represents operation
- Edge represents tensor
- CG is defined in high-level language
- Main purpose of CG is to compute gradients symbolically
- Separated execution and building



Kaggle Club
2017

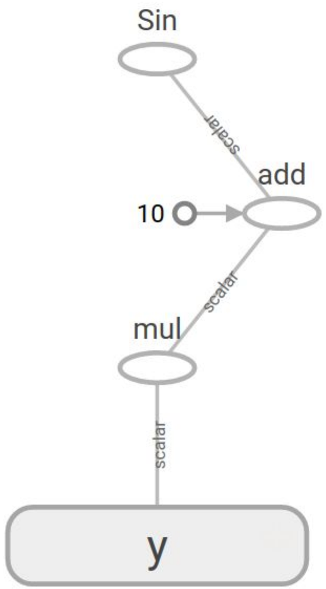
## Make Tensors Flow again!

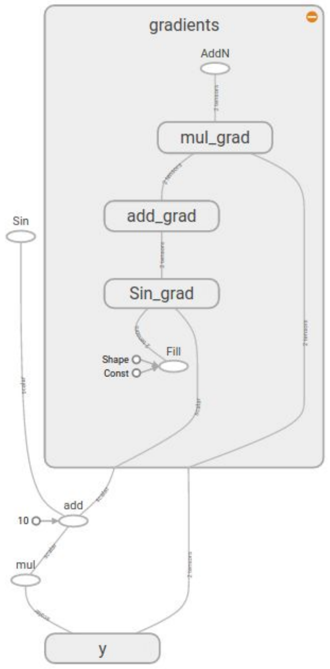
TF computational model

Consider following example:

$$f = \sin(y \cdot y + 10)$$

$$\frac{\partial f}{\partial y} = \cos(y \cdot y + 10) \cdot (y + y)$$





Kaggle Club
2017

## What’s in the box?

### What we should know in the beginning

4 sorts of tensors:

- tf.Variable (and non-trainable)
- tf.constant: immutable version of non-trainable variable
- tf.placeholder: is used for feeding training batches

Optimising functions:

- Many gradient descent optimizers: sgd, momentum, adagrad, adam, rmsprop, etc
- Also tf.hessians for quadratic optimization

All kind of mathematical operations:

- element-wise operations via standard arithmetic operators (*, +, -, etc) and functions (like tf.exp, tf.nn.softmax, etc)
- tensor operations (like tf.matmul, tf.nn.conv2d)

# Linear Models Example

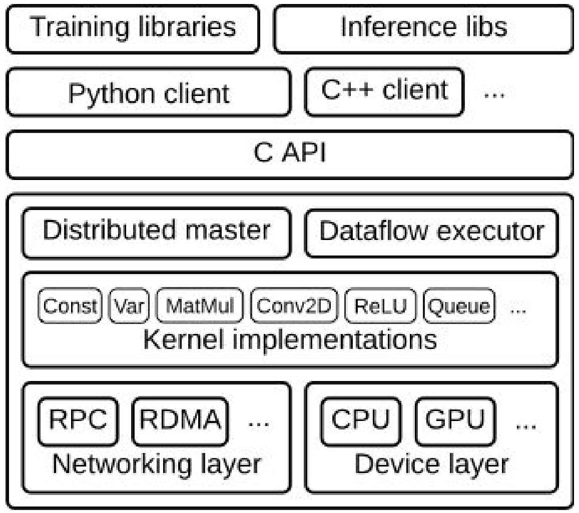
Kaggle Club
2017

## We need to go deeper

Let’s see how it works from inside

# Architecture

TensorFlow:
- Backend (written in C++ with CUDA)
- Frontends (Python, C++, Java, clients)



Frontend builds computational graphs, does the io stuff, and set up the environment.
Backend prunes a specific subgraph from the graph, as defined by the arguments to Session.run().
Distributed Master is for task management.

# TF Execution

Graphs does not provide execution of operations.
But session does.
That’s why graph is not a resource and session is.

While computing TF uses kernels. Kernel is specific pre-compiled function.
After building graph operation can be executed instantly by chaining all depending kernels.

Kernels for cpu is approximately common functions.
Kernel for gpu is CUDA kernel.
This means that if you want to create new operation you should implement it twice.

## Execution Example

## Visualization

TensorBoard - is not only image board

TensorBoard can visualize various things
- Scalars (usually it is loss curve or metric)
- Histograms
- Images
- Graphs

Everything defined in tf.summary e.g.
- tf.summary.scalar
- tf.summary.histogram
- tf.summary.image

TensorBoard can’t read all these data directly.
You should compute every summary object explicitly by passing to Session.run() and write summary with tf.summary.FileWriter

## Visualization Example

Kaggle Club
2017

## Saving Models

Use tf.train.Saver for saving weights.
Method .save collects all initialized variables in session and saves it into specified format.
Saver does not save graph.
Method .restore loads weights into the session.

Use tf.train.write_graph for saving graph in protobuf.
tf.train.import_graph_def loads graph from GraphDef object into default Graph object.

## Saving Example

Kaggle Club
2017

## High level API

- Code in pure TensorFlow completely unreadable.
- Even for simple model we need huge unreadable functions.

Now we consider a convolutional neural network model using high level tf.layers interface

We have a lot of libraries that abstract tf mechanisms and provide higher level blocks for constructing models

# Thanks!

Kaggle Club
2k17