TensorFlow

- Introduction -

2019 – 2020

Ando Ki, Ph.D.
adki@future-ds.com

Table of contents

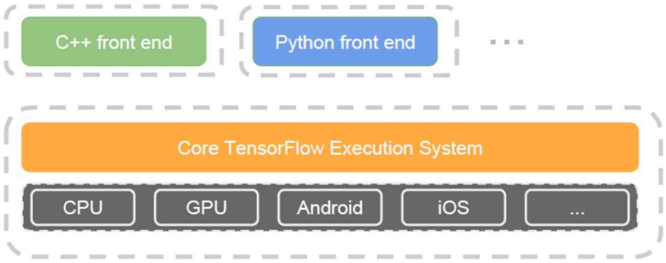
- Introduction to TensorFlow
  - Deep-Learning Package Design Choices
  - TensorFlow architecture
  - TensorFlow configuration
  - TensorFlow from Google
  - Tensorflow model

Deep-Learning Package Design Choices

- Model specification
  - Configuration file
    - e.g. Caffe, DistBelief, CNTK
  - Programmatic generation
    - Torch: Python
    - Theano: Python
    - TensorFlow: Python
    - Tiny-Dnn: C++
- For programmatic models, choice of high-level language:
  - Lua (Torch)
  - Python (Theano, TensorFlow)
    - Theano: Academic
    - TensorFlow: Google
  - C/C++
    - Caffe
    - Mxnet

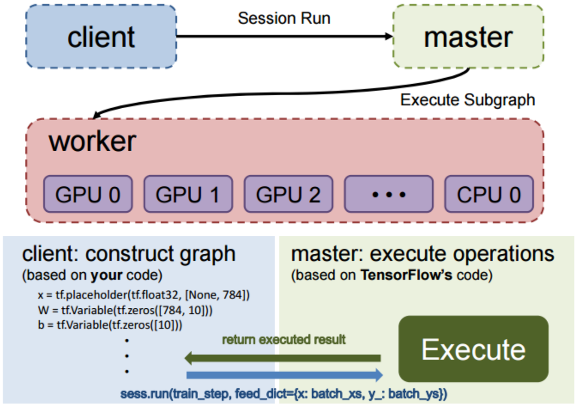
TensorFlow architecture

- Core in C++
  - Lowest level API (TensorFlow core): provide complete programming control
- Different front ends for specifying/driving the computation
  - Higher level API on top of the TensorFlow core
  - Python and C++



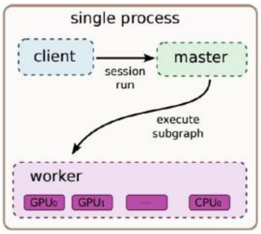
"TensorFlow: A System for Machine Learning on Heterogeneous Systems" by Jeff Dean / Google Brain team

TensorFlow configuration

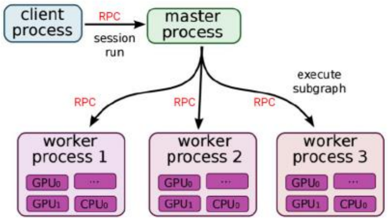


```
x = tf.placeholder(tf.float32, [None, 784])
W = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))
                  .
                  .
                  .
sess.run(train_step, feed_dict={x: batch_xs, y_: batch_ys})
```

- Single process configuration



- Distributed configuration

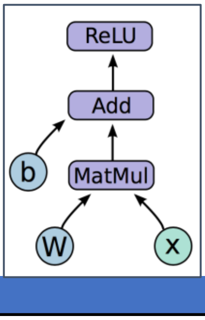


TensorFlow from Google

- TensorFlow is not a language
  - It is a framework which has algorithms for solving machine learning and deep learning problems.
- The core of TensorFlow is the dataflow graph representing computations.
  - TensorFlow is an open source library for numerical computation using data flow graphs
  - Nodes represent operations (ops), and
  - the edges represent tensors (multi-dimensional arrays, the backbone of TensorFlow).
  - The entire dataflow graph is a complete description of computations, which occur within a session, and are executed on devices (CPUs or GPUs).
- TensorFlow provides Python front-end
  - where tensors are represented internally as familiar numpy ndarray objects.
  - TensorFlow relies on highly-optimized C++ for its computation at its heart, i.e., TensoFlow core.
- The algorithmic engine is build over C++, on top of which there is a Python API acting as a bridge to call the C++ engine.
  - All operations are done outside of Python.
  - In short a Python programming interface to make life simpler for a developer so that he can readily use the underlying engine without worrying much about its internal intricacies.

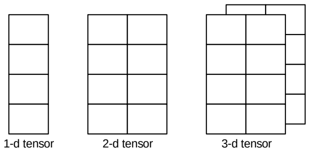
TensorFlow model

- Big idea
  - express a numeric computation as a graph, where tensors (data) flow through the graph
  - Graph: Data (Tensors) flow through the graph
    - Nodes: mathematical operations
    - edges: multi-dimensional arrays



Nodes: ReLU, Add, MatMul
Edges: b, W, x

- Tensor: N-dimensional array
  - 0-dimension: Scalar
  - 1-dimension: Vector (1-d tensor)
  - 2-dimension: Matrix (2-d tensor)
  - N-dimension: Tensor



- Flow: Computation based on data flow graphs
- Tensors flow through the graph
  - ➔TensorFlow
  - edges represent the tensors (data)
  - nodes represent the processing

㈜퓨쳐디자인시스템
34051 대전광역시 유성구 문지로 193, KAIST 문지캠퍼스, F723호
(042) 864-0211~0212 / contact@future-ds.com / www.future-ds.com

Future Design Systems, Inc.
Faculty Wing F723, KAIST Munji Campus, 193 Munji-ro, Yuseong-gu, Daejeon 34051, Korea
+82-042-864-0211~0212 / contact@future-ds.com / www.future-ds.com

FUTURE
Design Systems