Tensorflow

TensorFlow is an interface for expressing machine learning algorithms, and an implementation for executing such algorithms.

Outline

- Introduction
- VS.
- Basic concept about NN
- Programming model and basic concept
    - Operation and kernels
- Local vs. distributed system
- Example: MNIST
- Tensor Board
- Easy ML with tf.contrib.learn
- More advanced features
- Reference

Introduction

DistBelief

- Unsupervised learning
- Language representation
- Models for image classification and object detection
- Video classification
- Speech recognition
- Sequence prediction
- Move selection for Go
- Pedestrian detection
- Reinforcement learning
- And so on

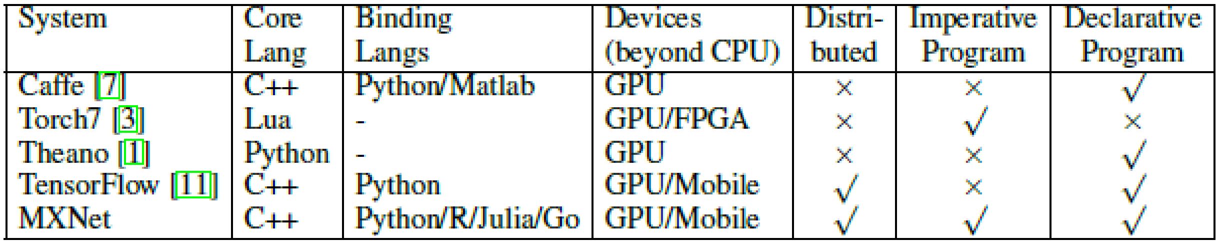
Used by:

- Google Search
- Advertising products
- Speech recognition systems
- Google Photos
- Google Maps and StreetView
- Google Translate
- YouTube
- And so on

Tensorflow is the second-generation system for the implementation and deployment of large-scale machine learning model.

- Programming model is more flexible.
- Performance is significantly better.
- Support training and using a broader range of models on a wider variety of heterogeneous hardware platforms

VS.



System	Core Lang	Binding Langs	Devices (beyond CPU)	Distri-buted	Imperative Program	Declarative Program
Caffe [7]	C++	Python/Matlab	GPU	×	×	√
Torch7 [3]	Lua	-	GPU/FPGA	×	√	×
Theano [1]	Python	-	GPU	×	×	√
TensorFlow [11]	C++	Python	GPU/Mobile	√	×	√
MXNet	C++	Python/R/Julia/Go	GPU/Mobile	√	√	√

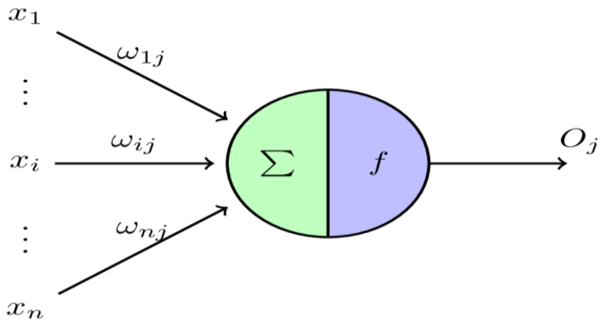
Pros/Cons of frameworks

https://deeplearning4j.org/compare-dl4j-torch7-pylearn

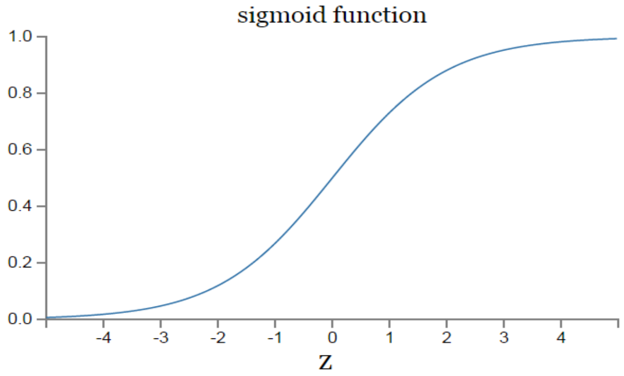
Basic concept about NN

Neuron:

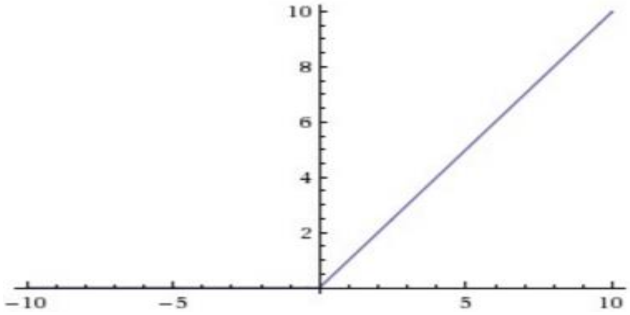
- A ‘neuron’ has a number of input connections
- Each input connection has a ‘weight’, that is multiplied by the value of the input
- The weighted inputs are then summed
- A ‘bias’ is then added to the sum of the weighted inputs
- This final ‘sum’ is then put through an ‘activation’ function to produce an output value



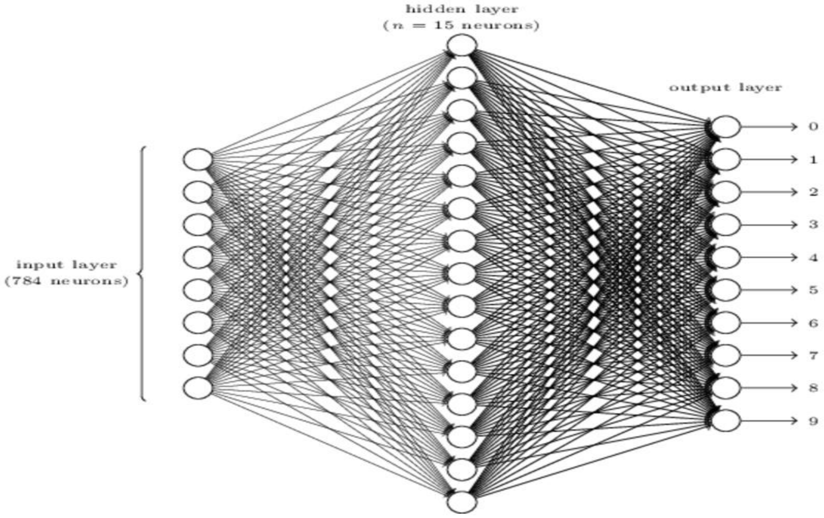
Activation Function:



ReLU max(0,x)



Layer:


hidden layer
(n = 15 neurons)
output layer
input layer
(784 neurons)
0
1
2
3
4
5
6
7
8
9

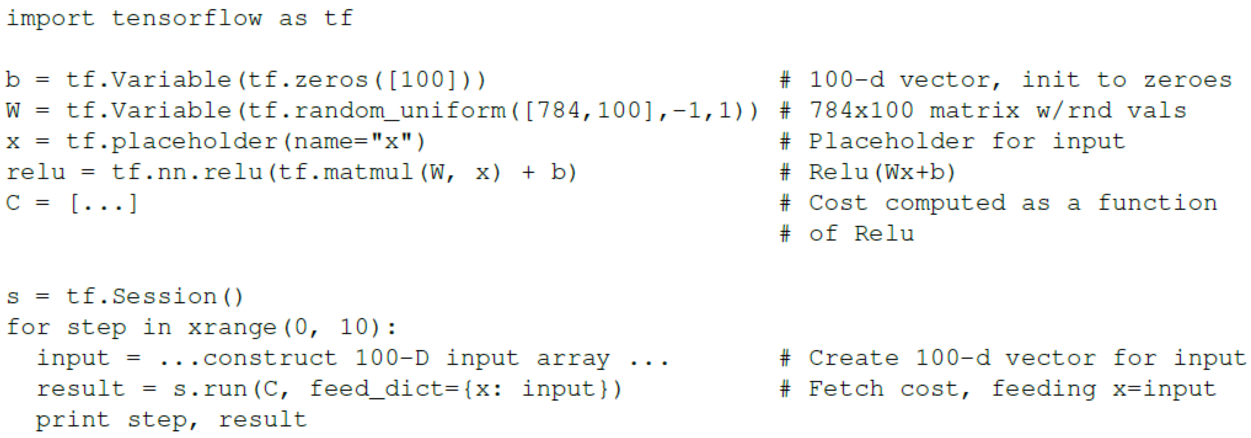
Programming Model and Basic Concepts

https://www.tensorflow.org/get_started/basic_usage

- Graph: represent a dataflow computation.
- Node: represent an operation (+ - * / …).
- Tensor: represent input/output of nodes.
- Variable: a special kind of operation that return a handle to a persistent mutable tensor.
- Session: client programs interact with the TensorFlow system by creating a Session.
- Client-Master: common concept for distributed system.

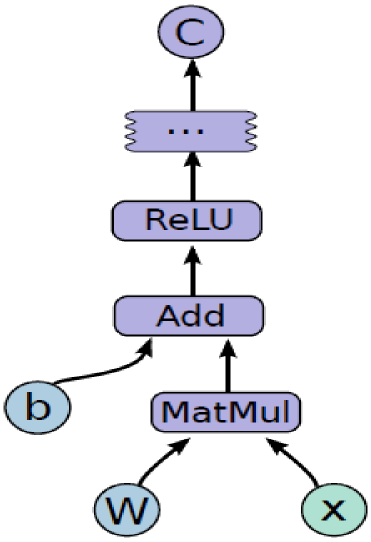
Client construct a computational graph using one of the supported frontend languages (C++ or Python). Call session.run() to submit the computation task.

Operations and Kernels

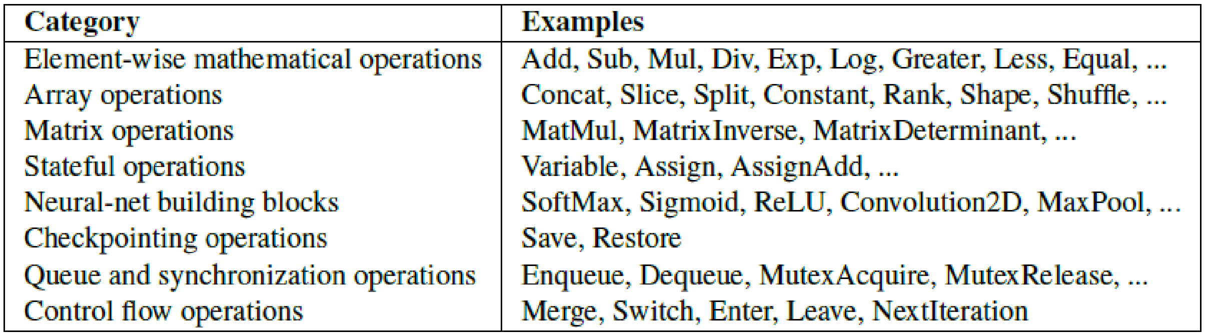


```
import tensorflow as tf

b = tf.Variable(tf.zeros([100]))                   # 100-d vector, init to zeroes
W = tf.Variable(tf.random_uniform([784,100],-1,1)) # 784x100 matrix w/rnd vals
x = tf.placeholder(name="x")                       # Placeholder for input
relu = tf.nn.relu(tf.matmul(W, x) + b)             # Relu(Wx+b)
C = [...]                                          # Cost computed as a function
                                                   # of Relu

s = tf.Session()
for step in xrange(0, 10):
  input = ...construct 100-D input array ...       # Create 100-d vector for input
  result = s.run(C, feed_dict={x: input})          # Fetch cost, feeding x=input
  print step, result
```

Figure 1: Example TensorFlow code fragment



Operation type



Category	Examples
Element-wise mathematical operations	Add, Sub, Mul, Div, Exp, Log, Greater, Less, Equal, ...
Array operations	Concat, Slice, Split, Constant, Rank, Shape, Shuffle, ...
Matrix operations	MatMul, MatrixInverse, MatrixDeterminant, ...
Stateful operations	Variable, Assign, AssignAdd, ...
Neural-net building blocks	SoftMax, Sigmoid, ReLU, Convolution2D, MaxPool, ...
Checkpointing operations	Save, Restore
Queue and synchronization operations	Enqueue, Dequeue, MutexAcquire, MutexRelease, ...
Control flow operations	Merge, Switch, Enter, Leave, NextIteration

Table 1: Example TensorFlow operation types

Local VS. distributed system

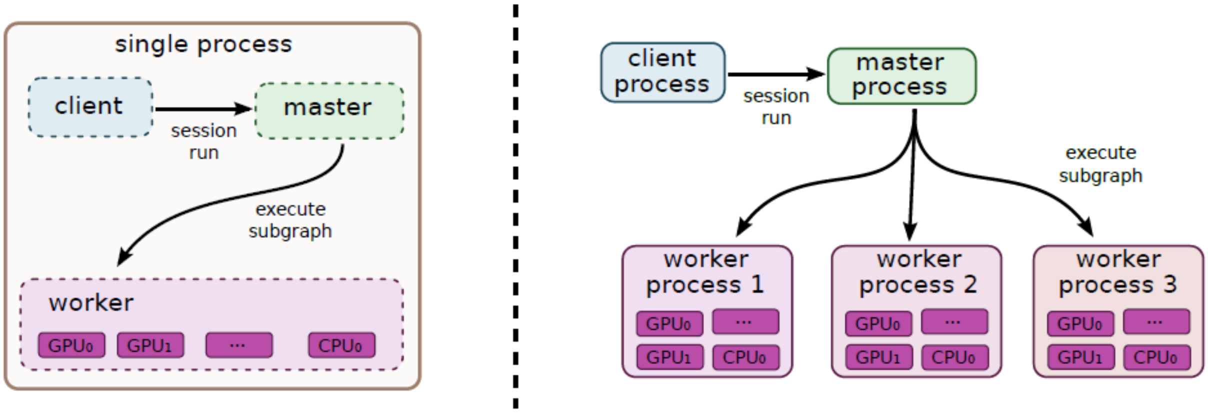
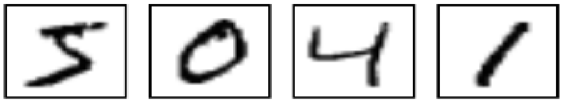


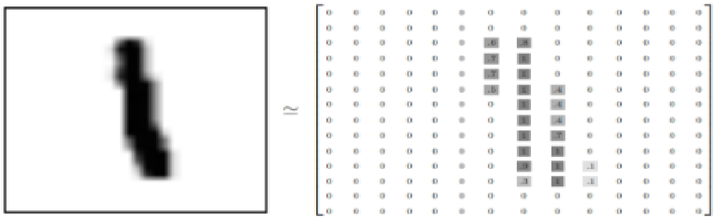
Figure 3: Single machine and distributed system structure

Example: MNIST

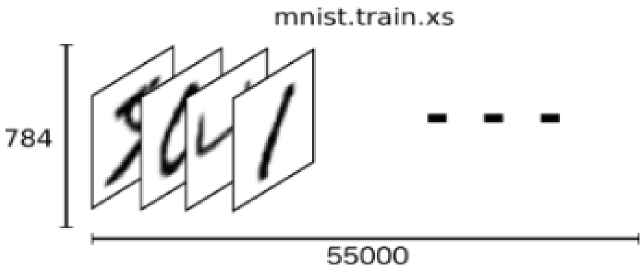
https://www.tensorflow.org/tutorials/mnist/beginners/

data source: 28x28

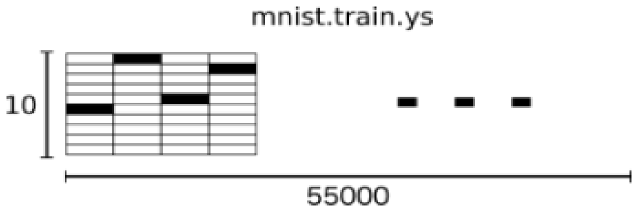




Input:

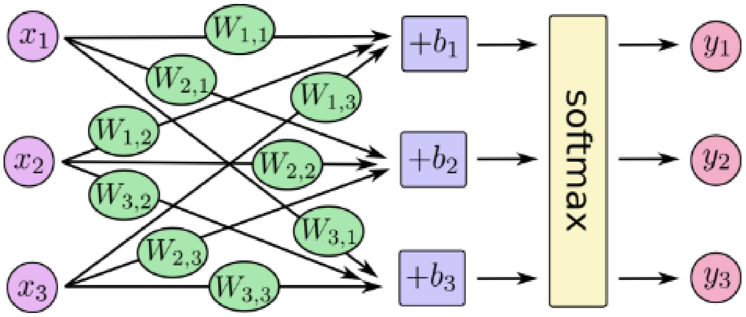


Target:



Softmax:

- ‘softmax’ is a function that takes N inputs, and produces N outputs whose values have properties of probability:
    - Sum to 1.0
    - Non-negative

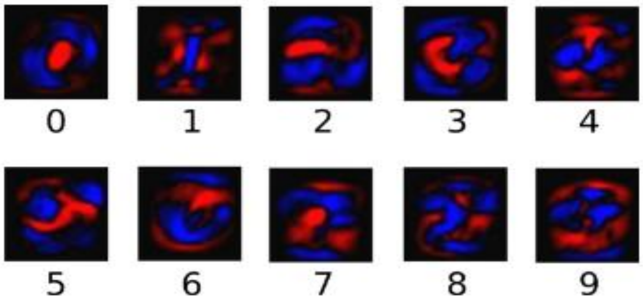


ations, we get:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax} \begin{pmatrix} W_{1,1}x_1 + W_{1,2}x_2 + W_{1,3}x_3 + b_1 \\ W_{2,1}x_1 + W_{2,2}x_2 + W_{2,3}x_3 + b_2 \\ W_{3,1}x_1 + W_{3,2}x_2 + W_{3,3}x_3 + b_3 \end{pmatrix}$$

cedure, turning it into a matrix multiplication and vector addition. This
It's also a useful way to think.)

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax} \left(\begin{bmatrix} W_{1,1} & W_{1,2} & W_{1,3} \\ W_{2,1} & W_{2,2} & W_{2,3} \\ W_{3,1} & W_{3,2} & W_{3,3} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right)$$



TensorBoard

A tool for graph visualization

https://www.tensorflow.org/how_tos/graph_viz/

Easy ML with tf.contrib.learn

TensorFlow’s high-level machine learning API.

- Load CSVs containing Iris training/test data into a TensorFlow Dataset
- Construct a neural network classifier
- Fit the model using the training data
- Evaluate the accuracy of the model
- Classify new samples

https://www.tensorflow.org/tutorials/tflearn/

Several more advanced features

- **Gradient Computation**
- **Partial Execution:** execute just a subgraph of the entire execution graph.
- **Device Constraints:** specify the device to execute one operation. (control the placement of nodes on devices)
- **Control Flow:** add control flow for graph. (if…else/do…while/…)
- **Input Operations:** worker read data from underlying storage system into memory directly.
- **Queues:** different portions of the graph can run asynchronously.
- **Containers:** Variable lives in containers.

Reference

Tensorflow tutorials: https://www.tensorflow.org/tutorials/

Tensorflow white paper: https://www.tensorflow.org/resources/

MxNet tutorials: http://mxnet.io/tutorials/index.html