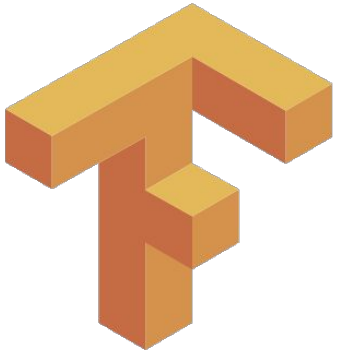
inovex

Let’s hack TensorFlow



Meetup

Karlsruhe, 8.6.2017

Zoran Cupic

Data Science Student

Alexandra Wörner

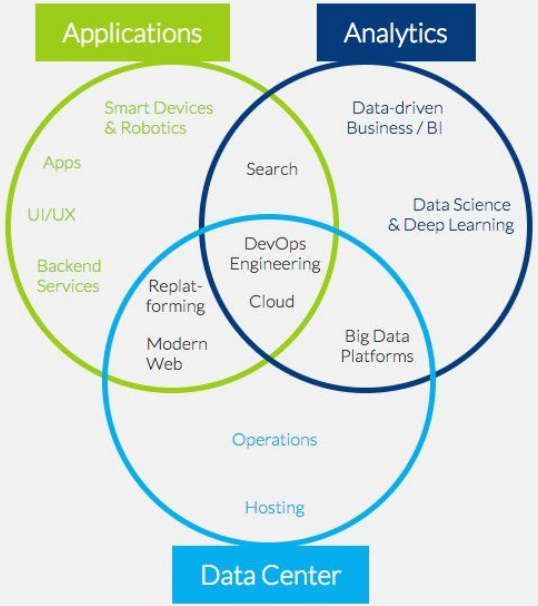
Big Data Scientist

Marcel Kurovski

Master's Degree Candidate

About inovex

- Enabling Your Digital Future
- about 250 inovexperts
- Offices: Karlsruhe, Pforzheim, Stuttgart, Munich, Cologne, Hamburg
- Data Science Projects
- Data Science Research at our inovex Lab

Strategisch	Technologisch	Methodisch
Data Products		Agile
Product Discovery		Digital Quality
Product Ownership		Training

inovex Lab

Feedback

https://sayat.me/tfinovexmeetup

Contents

Useful Links

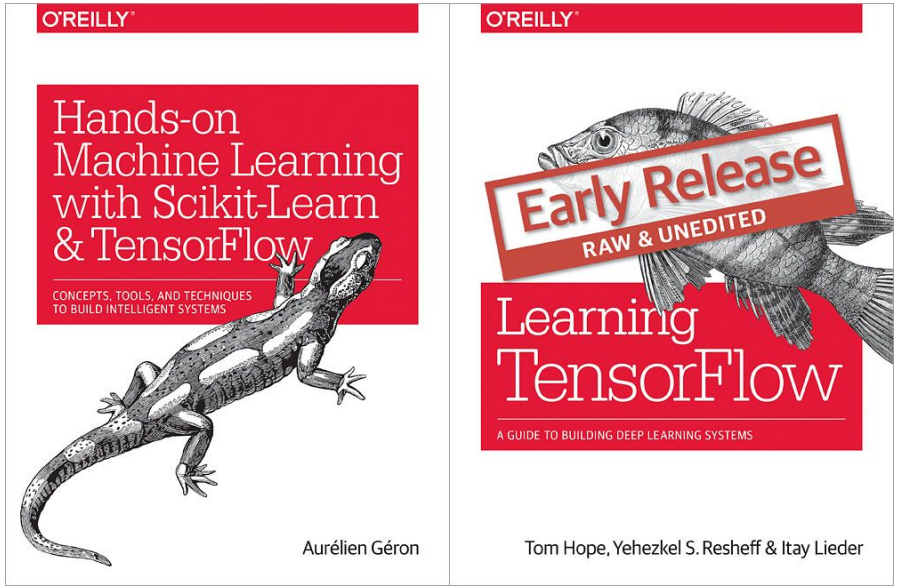
- Homepage: https://www.tensorflow.org/
- Documentation: https://www.tensorflow.org/api_docs/
- GitHub: https://github.com/tensorflow/tensorflow
- Model Zoo: https://github.com/tensorflow/models
- Awesome-Repo: https://github.com/jtoy/awesome-tensorflow/

Code Snippets

- provided as Jupyter notebooks
- Repository Link: *https://github.com/inovex/tensorflow-meetup*
- clone the repository
- run `jupyter notebook` from the repository root to start Jupyter

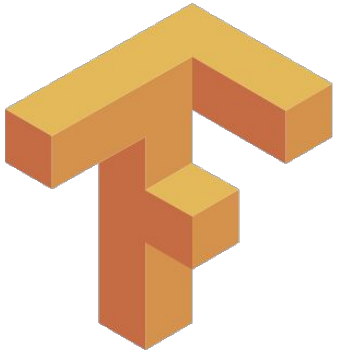
Literature

› lots of great stuff on the web!



inovex

Introduction to TensorFlow



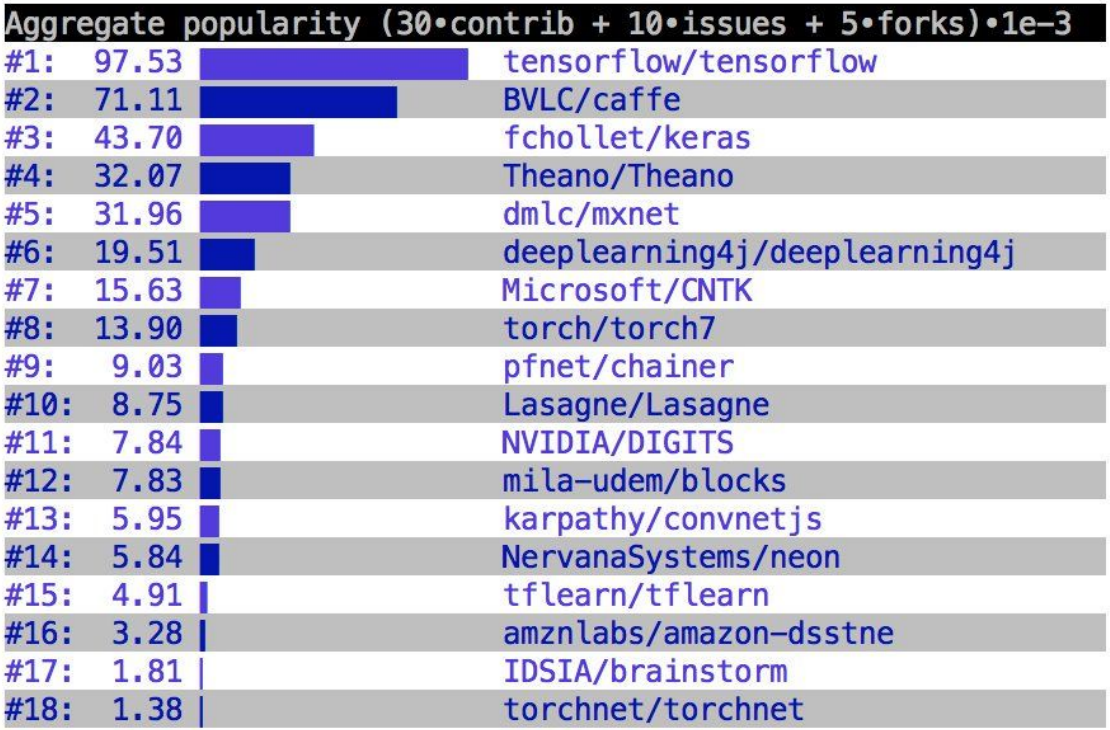
Introduction to TensorFlow

Overview

- open-source software library for Machine Intelligence
- developed by Google and open-sourced in Nov. 2015
- numerical computation using data flow graphs
- most used and most popular software library for Deep Learning

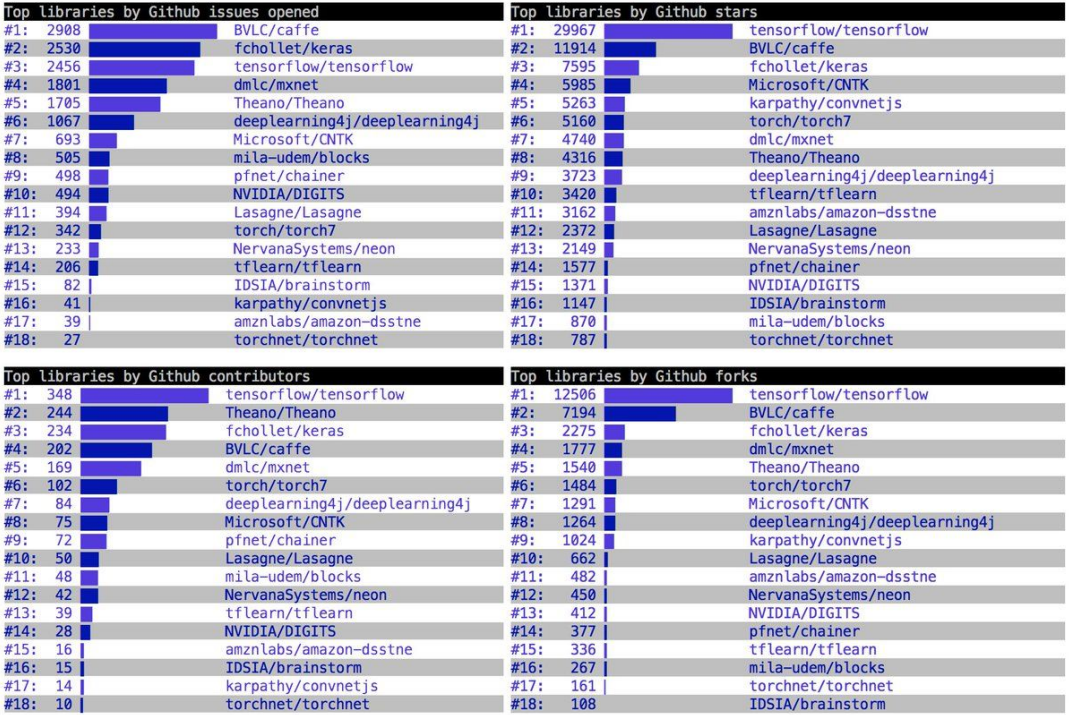
Introduction to TensorFlow

Overview



Introduction to TensorFlow

Overview



Top libraries by Github issues opened

#1:	2908	BVLC/caffe
#2:	2530	fchollet/keras
#3:	2456	tensorflow/tensorflow
#4:	1801	dmlc/mxnet
#5:	1705	Theano/Theano
#6:	1067	deeplearning4j/deeplearning4j
#7:	693	Microsoft/CNTK
#8:	505	mila-udem/blocks
#9:	498	pfnet/chainer
#10:	494	NVIDIA/DIGITS
#11:	394	Lasagne/Lasagne
#12:	342	torch/torch7
#13:	233	NervanaSystems/neon
#14:	206	tflearn/tflearn
#15:	82	IDSIA/brainstorm
#16:	41	karpathy/convnetjs
#17:	39	amznlabs/amazon-dsstne
#18:	27	torchnet/torchnet

Top libraries by Github stars

#1:	29967	tensorflow/tensorflow
#2:	11914	BVLC/caffe
#3:	7595	fchollet/keras
#4:	5985	Microsoft/CNTK
#5:	5263	karpathy/convnetjs
#6:	5160	torch/torch7
#7:	4740	dmlc/mxnet
#8:	4316	Theano/Theano
#9:	3723	deeplearning4j/deeplearning4j
#10:	3420	tflearn/tflearn
#11:	3162	amznlabs/amazon-dsstne
#12:	2372	Lasagne/Lasagne
#13:	2149	NervanaSystems/neon
#14:	1577	pfnet/chainer
#15:	1371	NVIDIA/DIGITS
#16:	1147	IDSIA/brainstorm
#17:	870	mila-udem/blocks
#18:	787	torchnet/torchnet

Top libraries by Github contributors

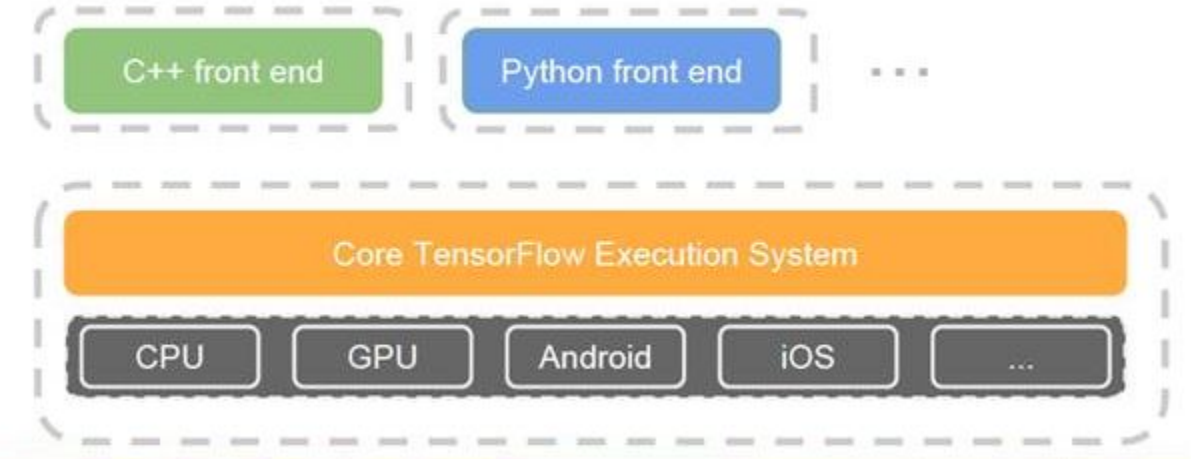
#1:	348	tensorflow/tensorflow
#2:	244	Theano/Theano
#3:	234	fchollet/keras
#4:	202	BVLC/caffe
#5:	169	dmlc/mxnet
#6:	102	torch/torch7
#7:	84	deeplearning4j/deeplearning4j
#8:	75	Microsoft/CNTK
#9:	72	pfnet/chainer
#10:	50	Lasagne/Lasagne
#11:	48	mila-udem/blocks
#12:	42	NervanaSystems/neon
#13:	39	tflearn/tflearn
#14:	28	NVIDIA/DIGITS
#15:	16	amznlabs/amazon-dsstne
#16:	15	IDSIA/brainstorm
#17:	14	karpathy/convnetjs
#18:	10	torchnet/torchnet

Top libraries by Github forks

#1:	12506	tensorflow/tensorflow
#2:	7194	BVLC/caffe
#3:	2275	fchollet/keras
#4:	1777	dmlc/mxnet
#5:	1540	Theano/Theano
#6:	1484	torch/torch7
#7:	1291	Microsoft/CNTK
#8:	1264	deeplearning4j/deeplearning4j
#9:	1024	karpathy/convnetjs
#10:	662	Lasagne/Lasagne
#11:	482	amznlabs/amazon-dsstne
#12:	450	NervanaSystems/neon
#13:	412	NVIDIA/DIGITS
#14:	377	pfnet/chainer
#15:	336	tflearn/tflearn
#16:	267	mila-udem/blocks
#17:	161	torchnet/torchnet
#18:	108	IDSIA/brainstorm

Introduction to TensorFlow

Architecture



Introduction to TensorFlow

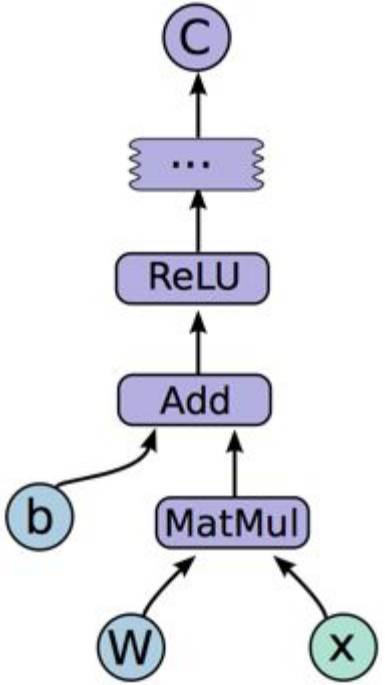
Programming Model & Basic Concepts

- Data Flow Graph
- Tensor
- Operations
- Variables
- Placeholder
- Session

Introduction to TensorFlow

Programming Model & Basic Concepts - Data Flow Graph

- computations represented as directed graph
  - Nodes: operations performed on data
  - Edges: flow of data / data → tensors
- 2 phases:
  - construction → assembling of graph
  - execution → run the graph with data



Introduction to TensorFlow

Programming Model & Basic Concepts - Tensor

- multidimensional array/matrix
  - scalar
  - vector
  - matrix
- tensors flow between nodes of the data flow graph

Introduction to TensorFlow

Programming Model & Basic Concepts - Operations

› computations performed in the data flow graph

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

Introduction to TensorFlow

Programming Model & Basic Concepts - Variables

- persistent mutable tensors
- a variable survives across multiple executions of a graph
- e. g. parameters of a machine learning model are variables
- alongside the variables there are also constants
  - persistent non-mutable tensors with a fixed value

Introduction to TensorFlow

Programming Model & Basic Concepts - Placeholder

- persistent mutable tensors
- fed with data at the execution of a graph
- e. g. data of a machine learning model

Introduction to TensorFlow

Programming Model & Basic Concepts - Session

- session allows the execution of a graph or parts of a graph
  - execution of defined computations → “run the graph”
- allocation of resources for execution
- data “lives” in the session

Introduction to TensorFlow

Programming Model & Basic Concepts

how it is done in practice:

- define everything you need and the logic
- create a session
- initialize all variables
- run the graph for **n** times

Introduction to TensorFlow

Programming Model & Basic Concepts - Example

```
import tensorflow as tf
import numpy as np

# define a variable, placeholder and a constant
z = tf.Variable(tf.random_normal([10,10], stddev=0.4, dtype=tf.float32), name="y")  # 10x10 tensor matrix
x = tf.placeholder(tf.float32, shape=(10, 10), name="x")  # Placeholder for input
b = tf.constant(2.5, dtype=tf.float32)

# perform  y = x*z+b
y = tf.matmul(x, z)
y = tf.add(y,b)
```

Introduction to TensorFlow

Programming Model & Basic Concepts - Example

```
# define session & initialize variables
sess = tf.Session()
init_op = tf.global_variables_initializer()
sess.run(init_op)

# run the graph 8 times in a loop and feed it each time with new data
for i in range(8):
  print("step: "+str(i+1))
  random_array = np.random.rand(10,10)
  evaled_z = sess.run(z,feed_dict={x: random_array})
  print(evaled_z)
```

to put it all together…

01_1_run_tensorflow.ipynb

enough with the basics, let’s start with some simple machine learning

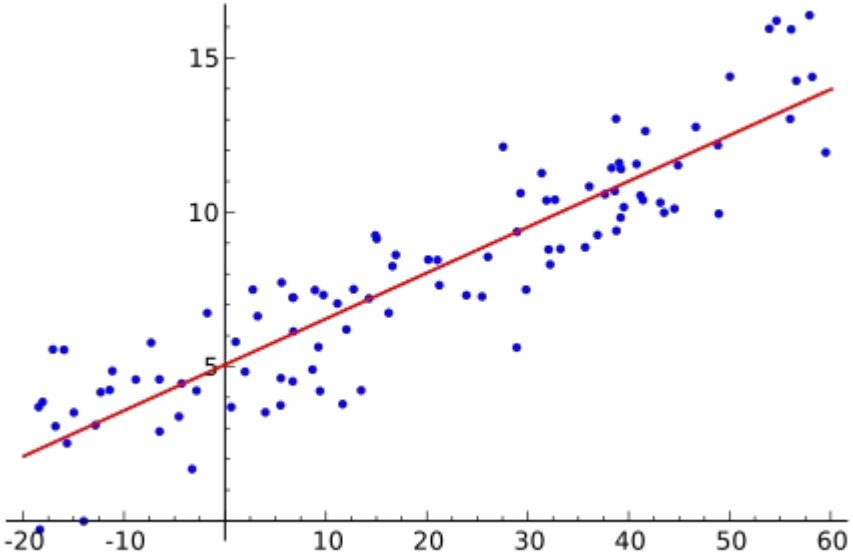
Introduction to TensorFlow

Linear Regression

- one of the simplest models → find a linear function
- correlation between a numeric variable /numeric variables and a numeric label
- find a function that fits best to the data points

Introduction to TensorFlow

Linear Regression



https://en.wikipedia.org/wiki/Linear_regression#/media/File:Linear_regression.svg

Introduction to TensorFlow

Linear Regression

- **y = x * w + b**
- x → vector with the input variable / variables
- w → vector with coefficient / coefficients
- b → constant

parameters

- function that fits best
  - → find best parameters

Introduction to TensorFlow

Linear Regression

- how to find the function with the best parameters ?
  - *optimize the parameters iteratively*

- optimization step:
  - *quantify the error with a error function*
  - *optimize the parameters by reference to the given error*

Introduction to TensorFlow

Linear Regression

- do so many optimization steps till the error is minimal
  - best parameters → function that fits best

Gradient Descent

https://en.wikipedia.org/wiki/Gradient_descent

https://spin.atomicobject.com/2014/06/24/gradient-descent-linear-regression/

Introduction to TensorFlow

Linear Regression in TensorFlow

01_2_linear_regression.ipynb