



# TensorFlow

powered by: python



by M. Behzadi



# Introduction

# What is Tensorflow?

- ✓ **Open-source**
- ✓ **Library**
- ✓ **Created by Google Brain Team**
- ✓ **Numerical Computation**
- ✓ **Machine Learning**
- ✓ **Deep Learning**
- ✓ **High Performance** (C++)
- ✓ **Convenient front-end API**  
(Python, JavaScript, C++, Java, Go, C# and Julia)
- ✓ **CPU/GPU/TPU Support**

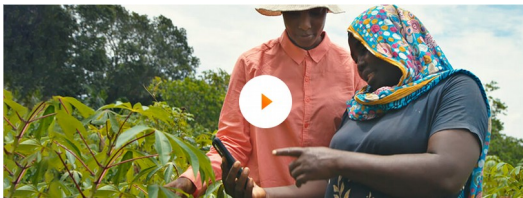
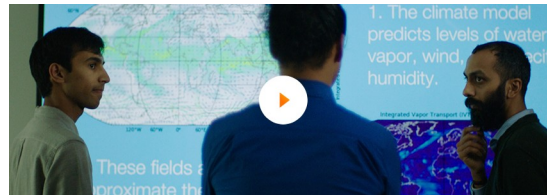
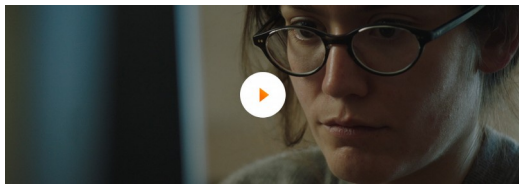


# History

- ✓ **Created by Google Brain Team**
- ✓ **Was part of a Google Product named DistBelief**
- ✓ **Open Sourced in late 2015**
- ✓ **Under Apache License**  
(Use, Modify, Redistribute, Sell)
- ✓ **First Stable version in 2017**
- ✓ **Latest Release: Tensorflow 1.13.2**
- ✓ **TensorFlow 2.0 RC released on Aug 23**

# Who is using Tensorflow?

- ✓ Researchers
- ✓ Data Scientists
- ✓ Programmers



CEVA

swisscom

SINO VATION  
VENTURES

中国移动  
China Mobile

carousell

WPS  
Qualcomm

Lenovo

Coca-Cola

Google



airbnb



AIRBUS  
DEFENCE & SPACE

arm

GE Healthcare

intel

PayPal

# Where can Tensorflow run?

## ✓ **Development Phase (Training)**

- Your Desktop or laptop
- Multiple machines

## ✓ **Run Phase (Test)**

- Desktop running Windows, macOS or Linux
- Cloud as a web service
- Mobile devices like iOS and Android

# Language Support

- ✓ **C++ as TF core**
- ✓ **Python 2.7/3.4-3.7**
- ✓ **JavaScript (tensorflow.js)**
- ✓ **Rust**
- ✓ **R**
- ✓ **Golang**
- ✓ **Java**
- ✓ **C#**
- ✓ **Julia**

# Other Rivals





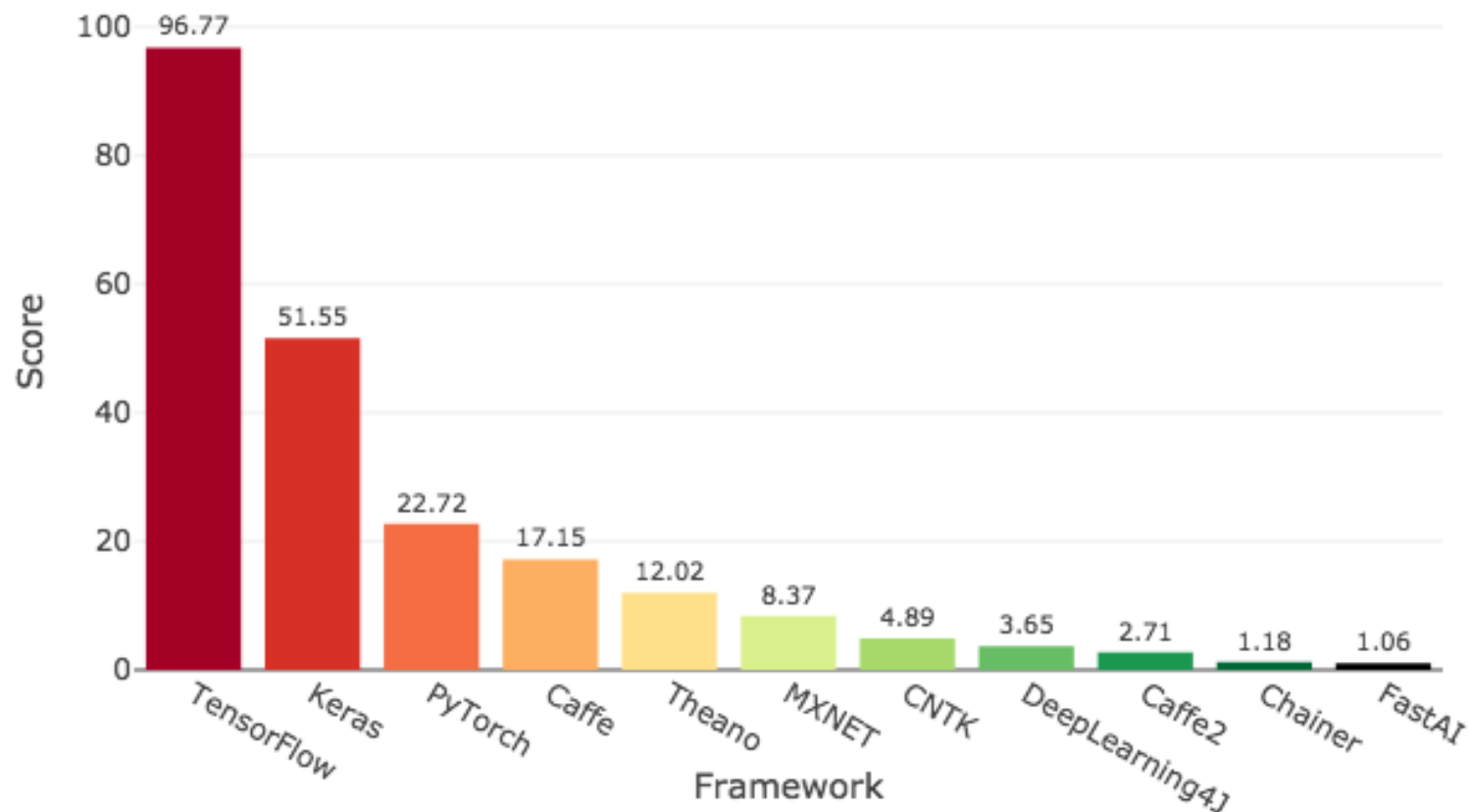
# Deep Learning Framework Power Scores (2018)

- ✓ **Kaggle Kernel**
- ✓ **Article**



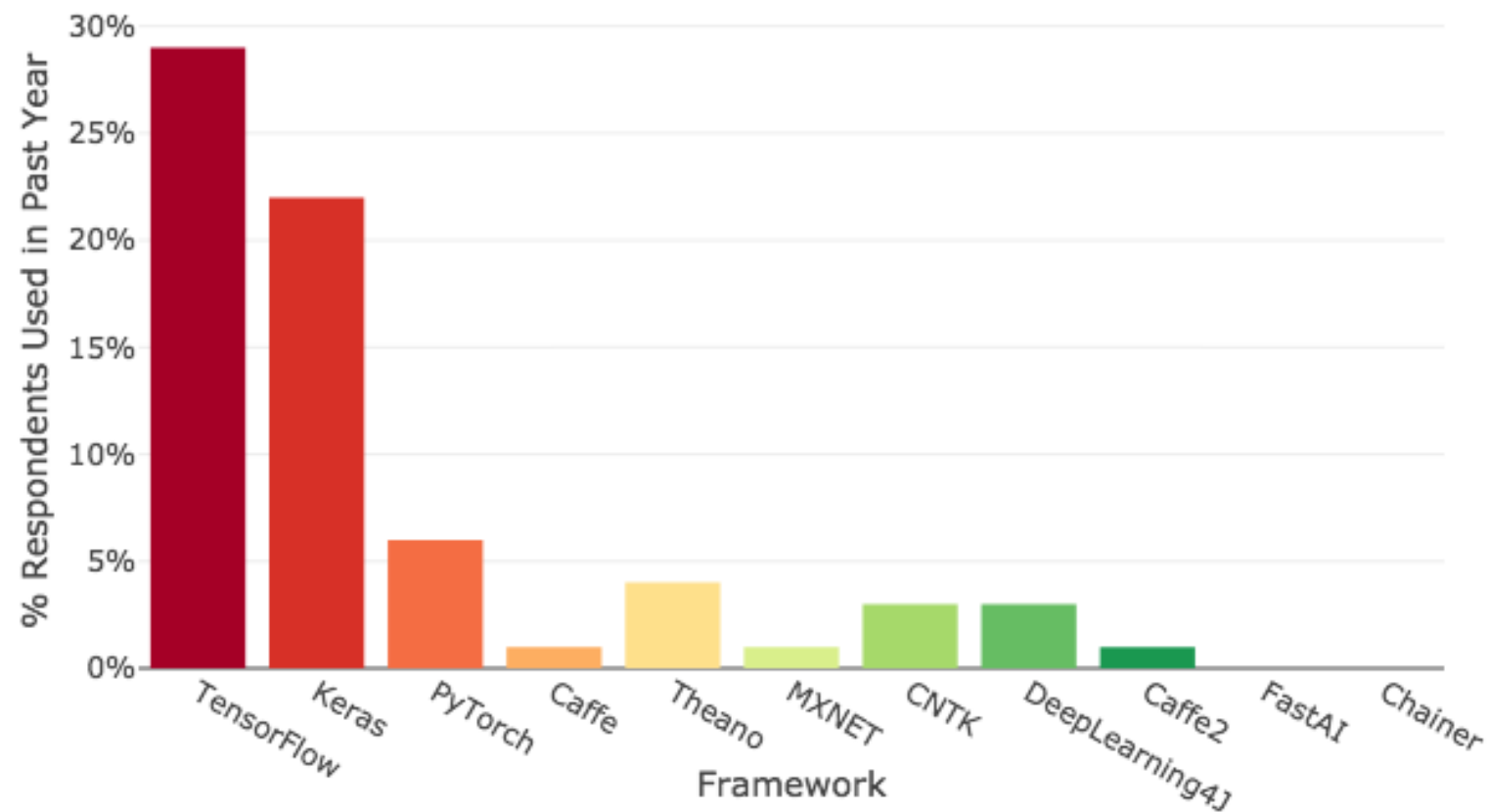
# Comparison (2018)

Deep Learning Framework Power Scores 2018

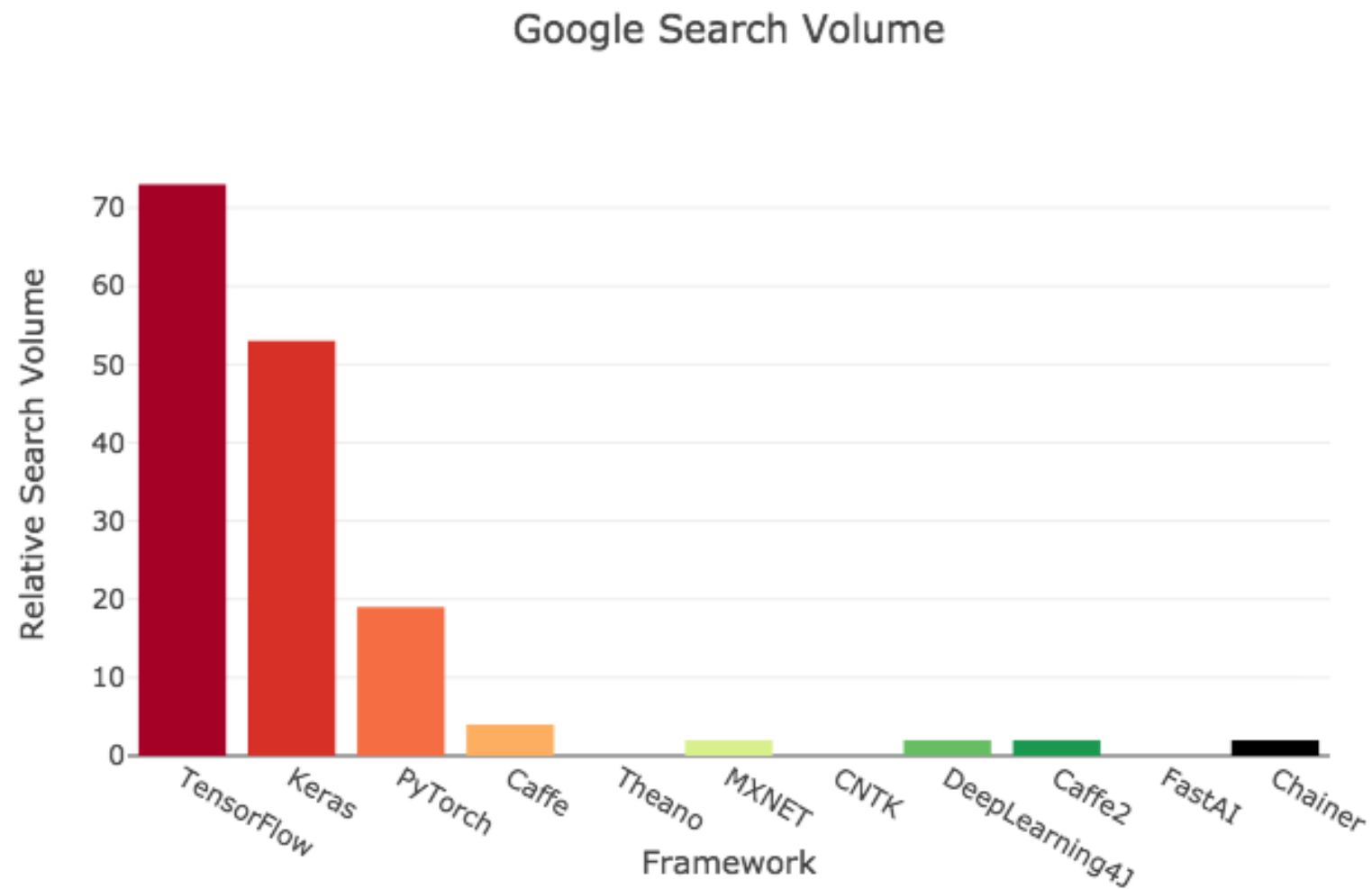


# Comparison (2018)

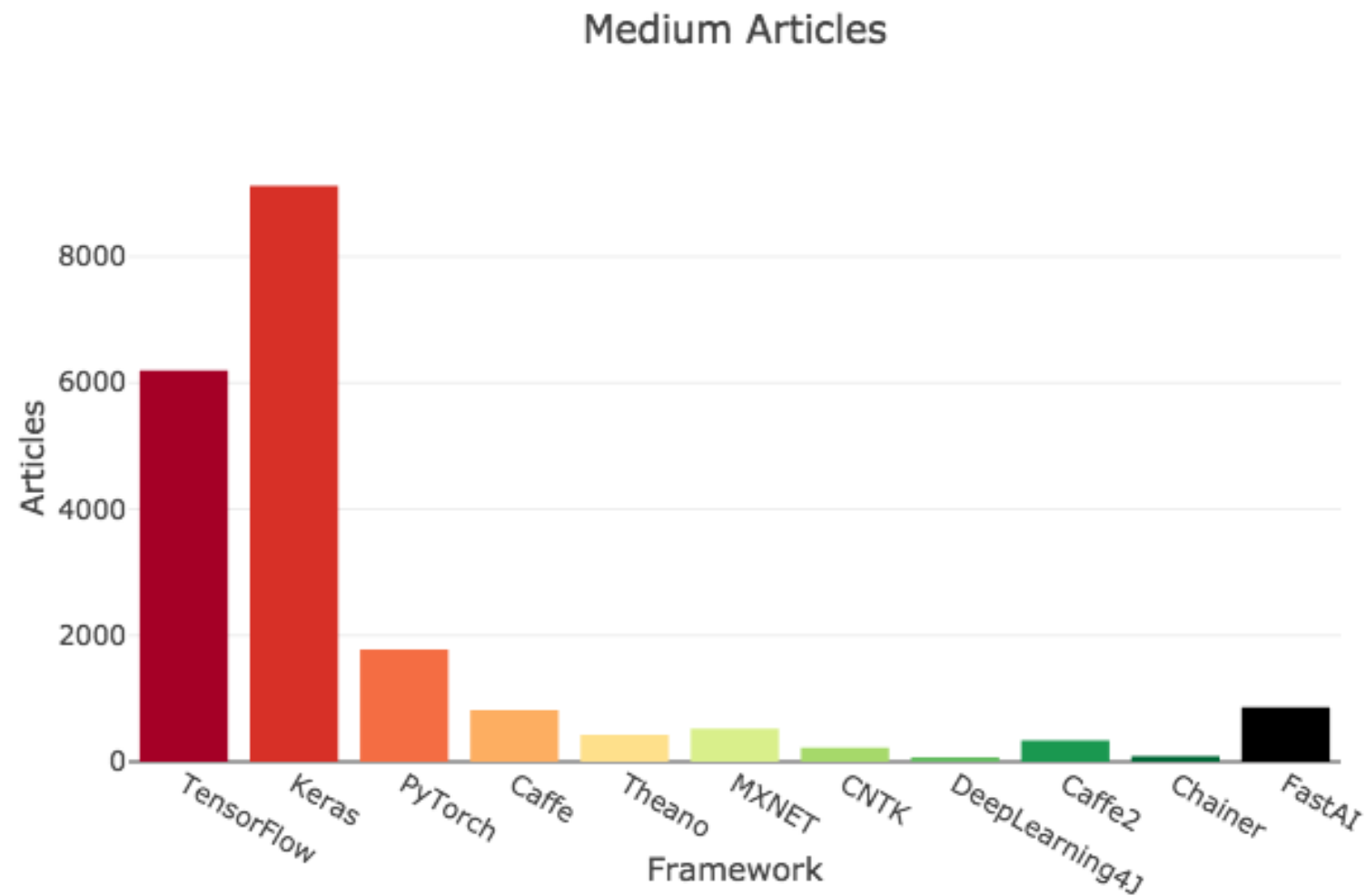
KDnuggets Usage Survey



# Comparison (2018)



# Comparison (2018)

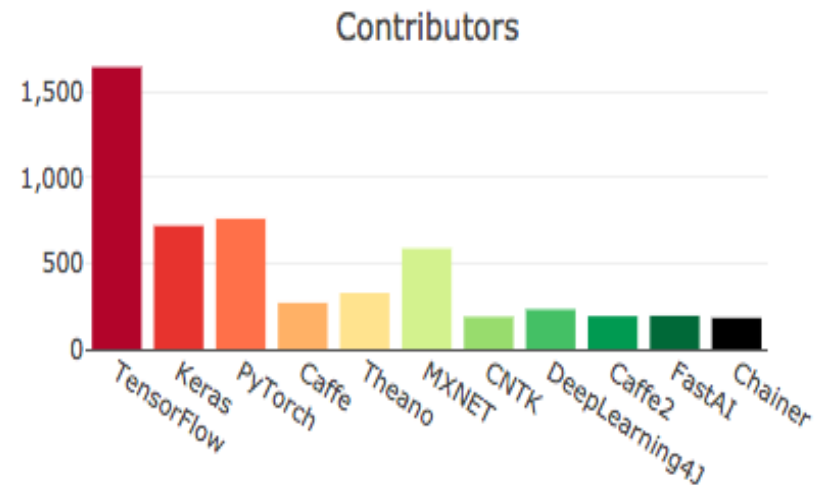
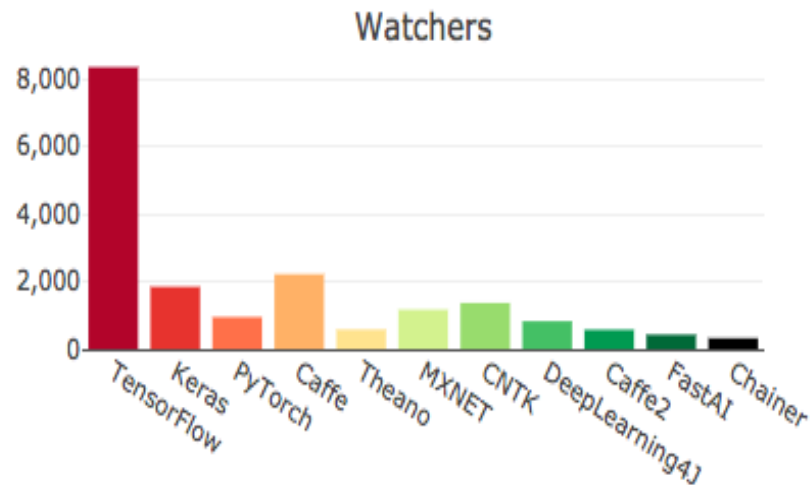
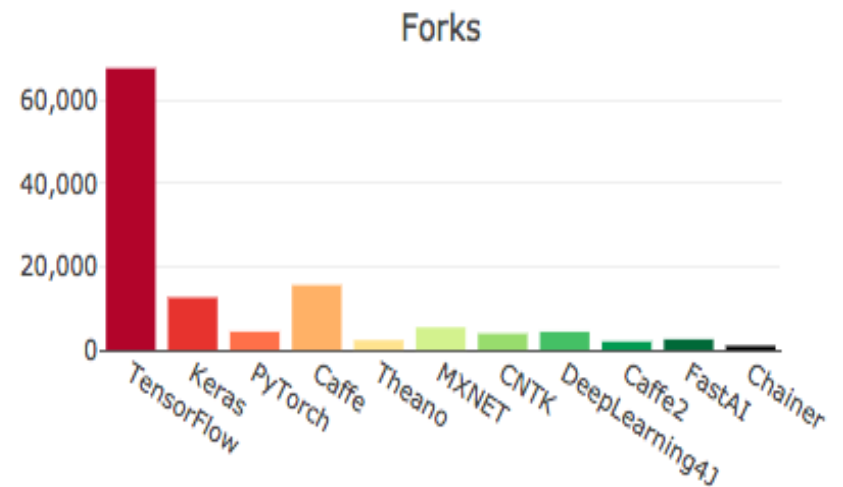
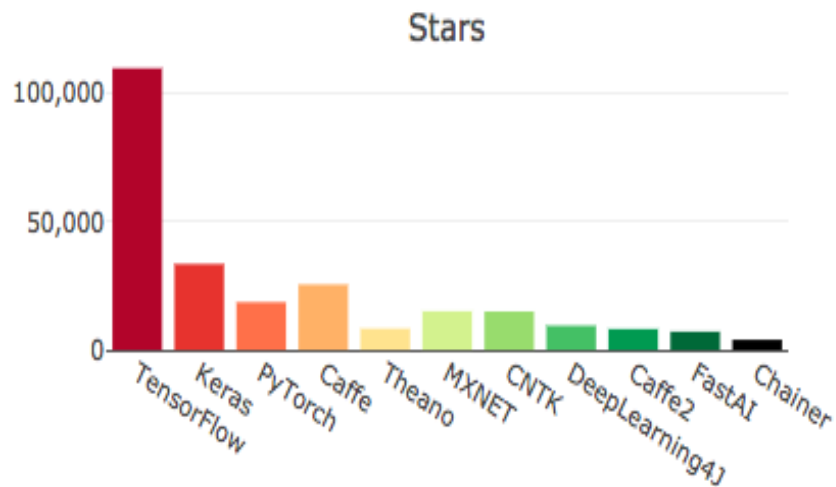


# Comparison (2018)

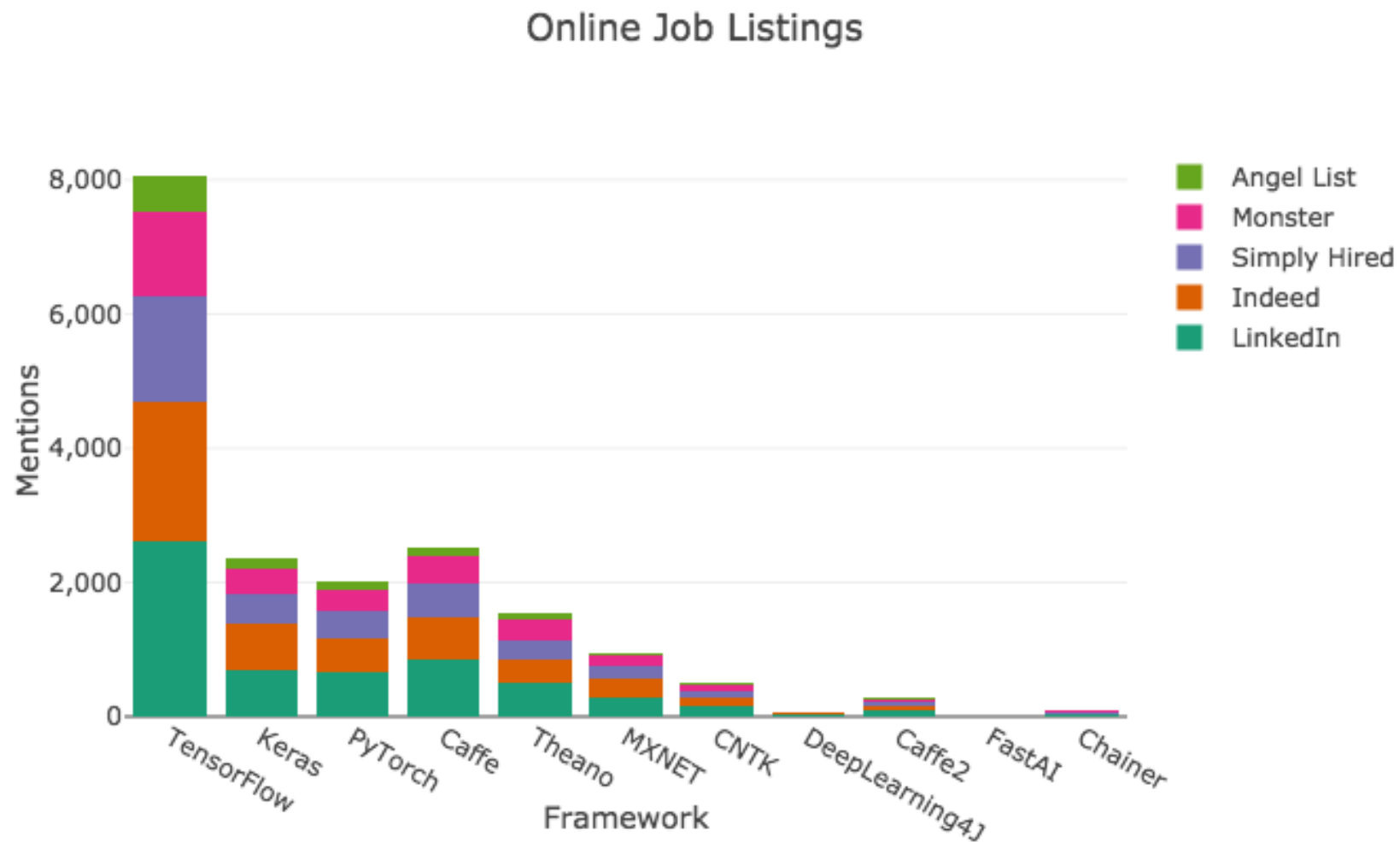


# Comparison (2018)

GitHub Activity



# Comparison (2018)



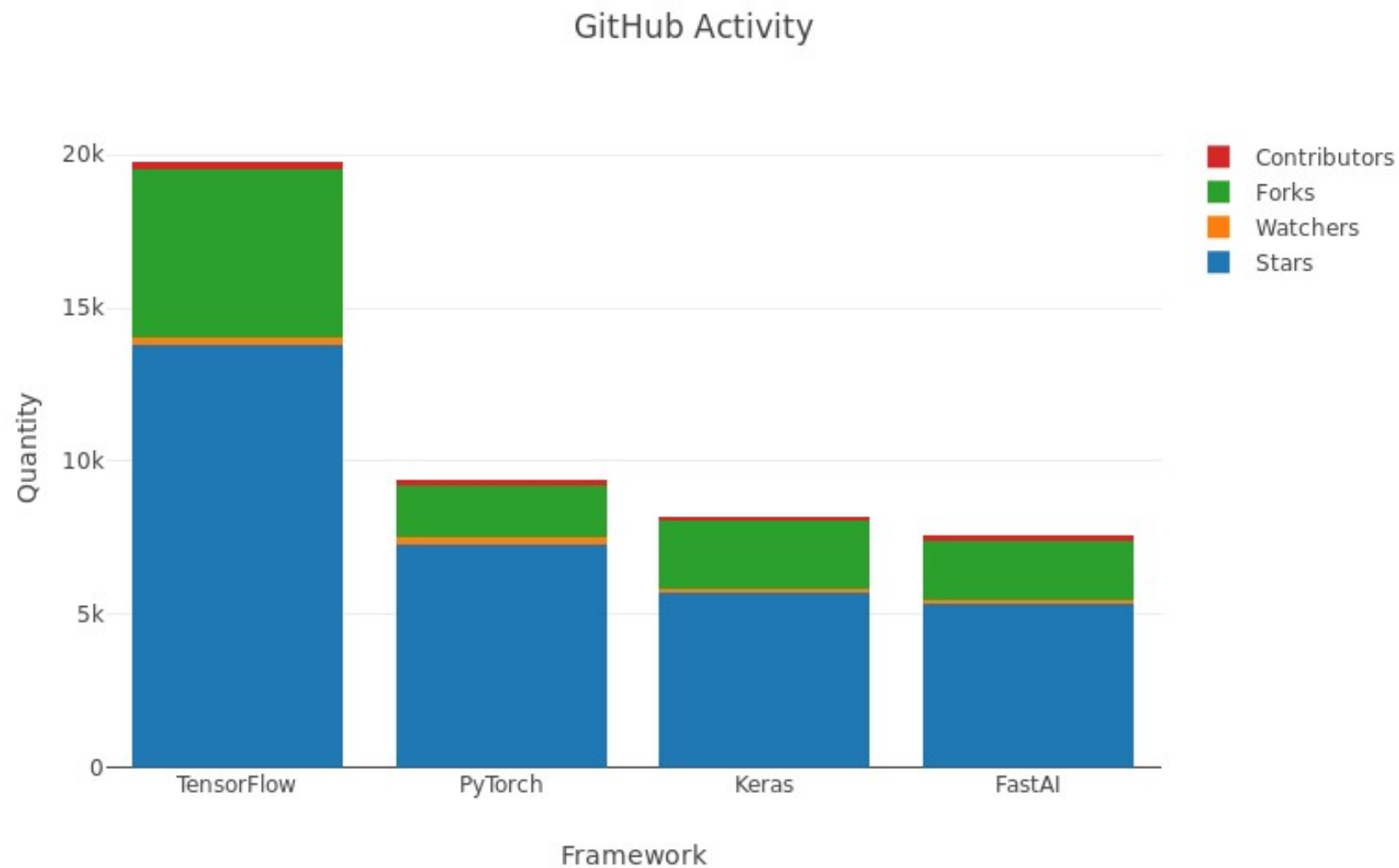


# Which Deep Learning Framework is Growing Fastest? (2019)

- ✓ **Kaggle Kernel**
- ✓ **Article**



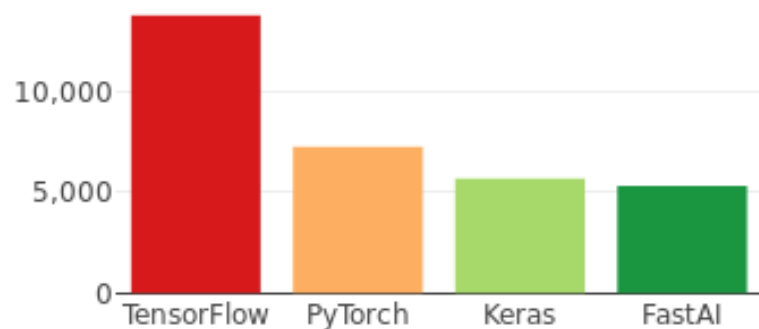
# Comparison (2019)



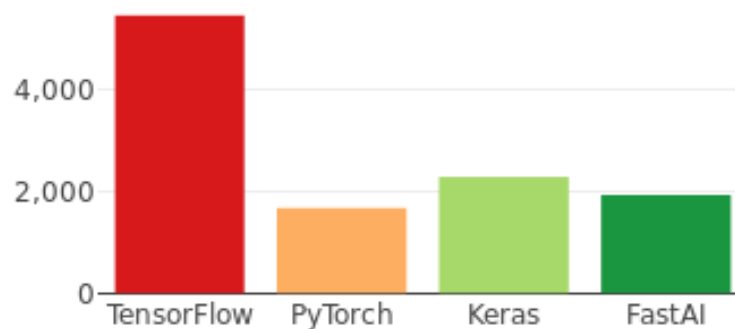
# Comparison (2019)

New GitHub Activity

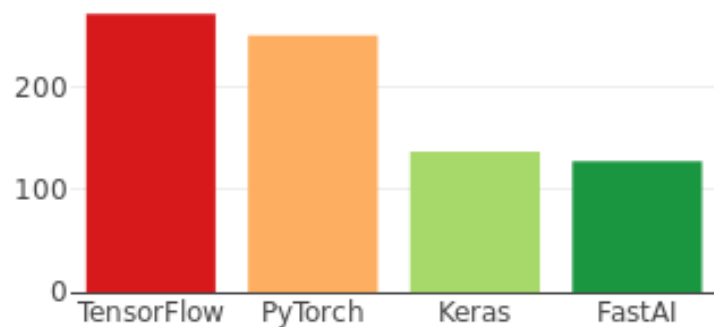
Stars



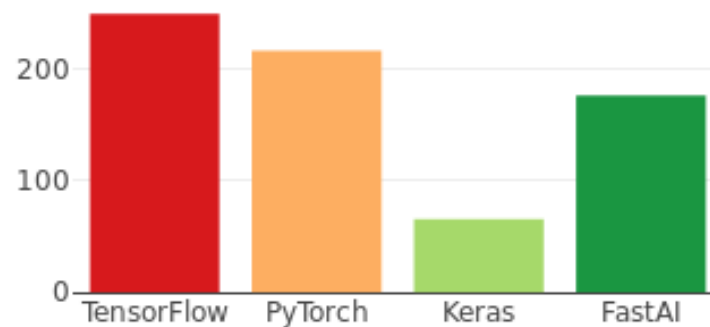
Forks



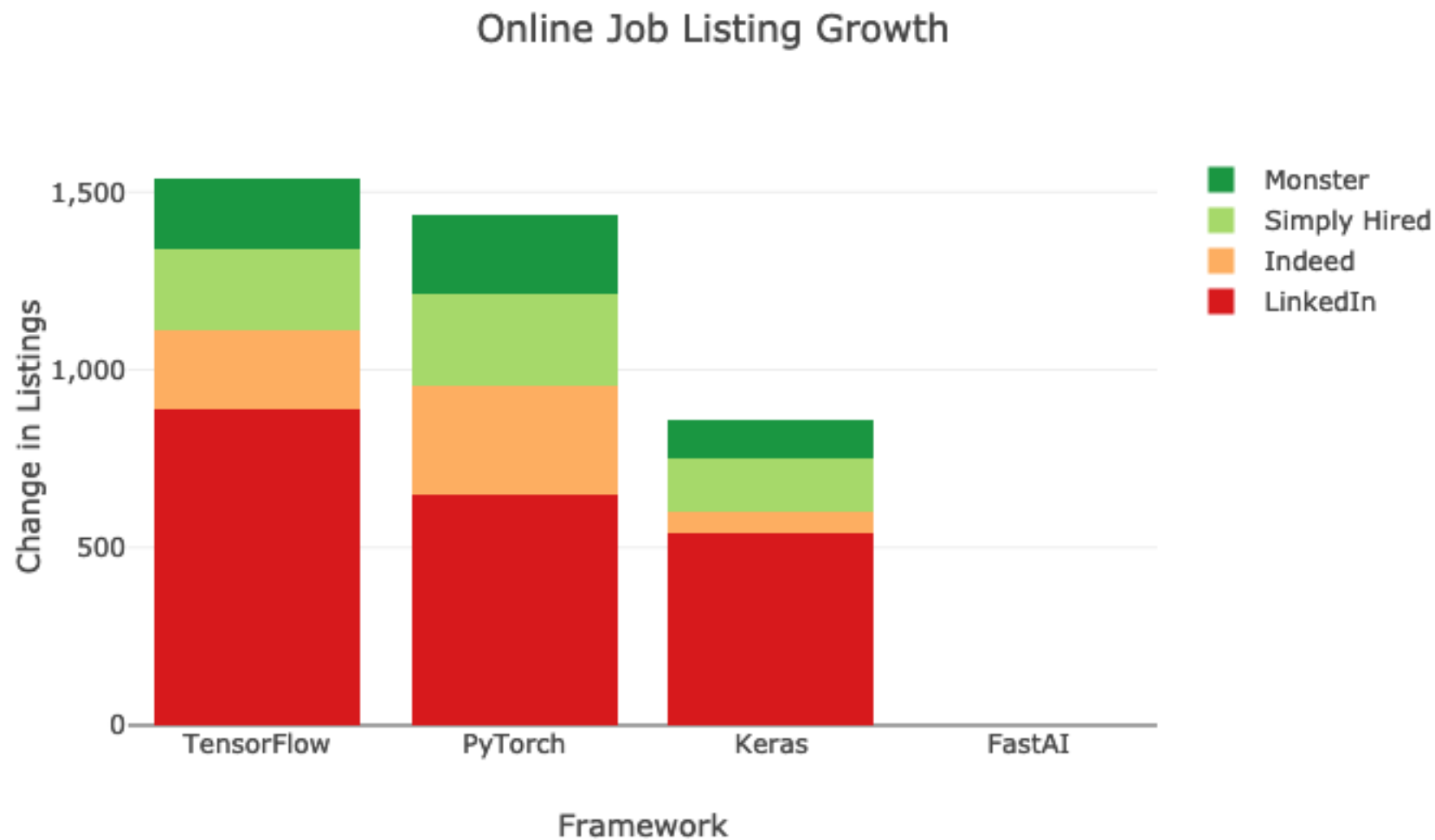
Watchers



Contributors

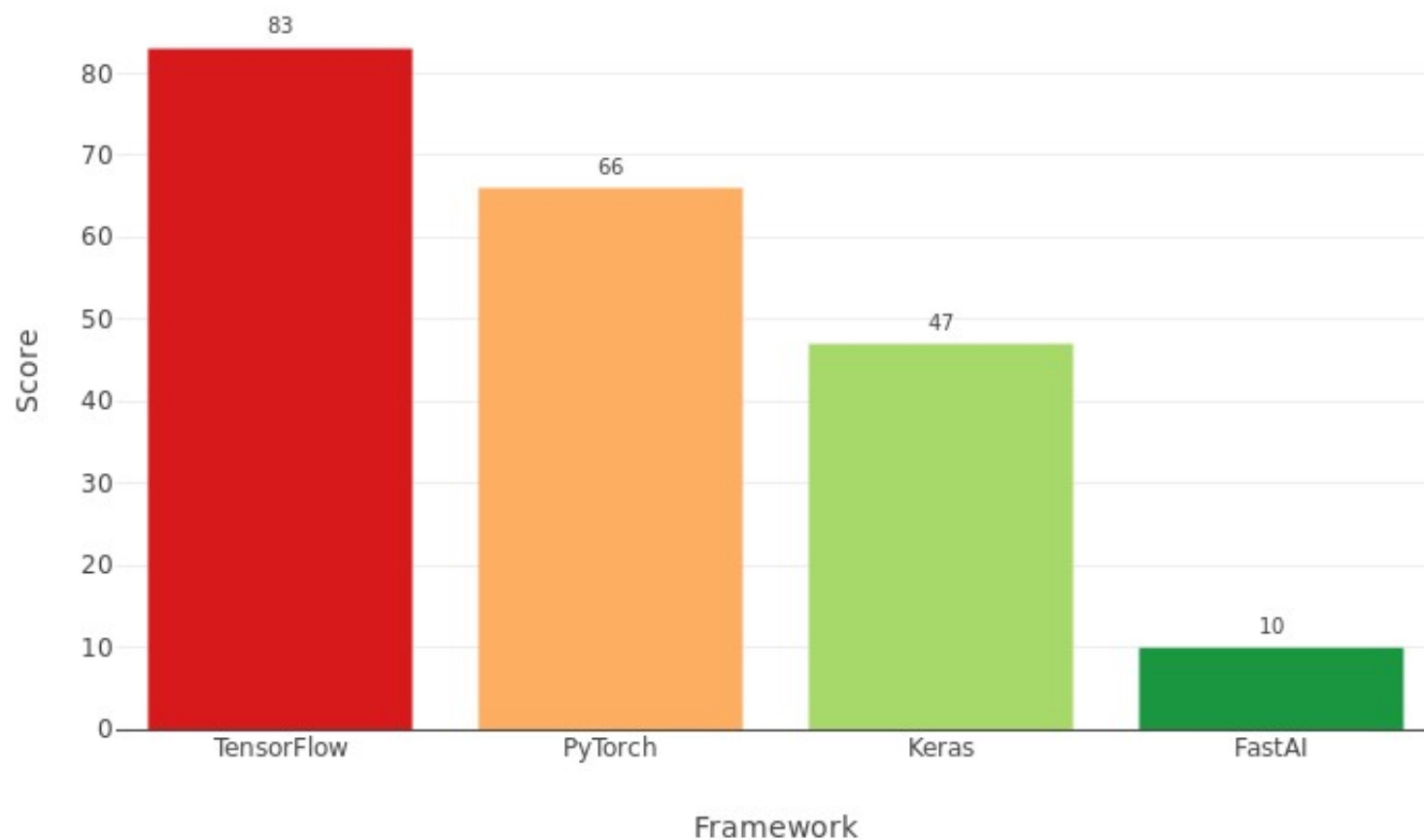


# Comparison (2019)

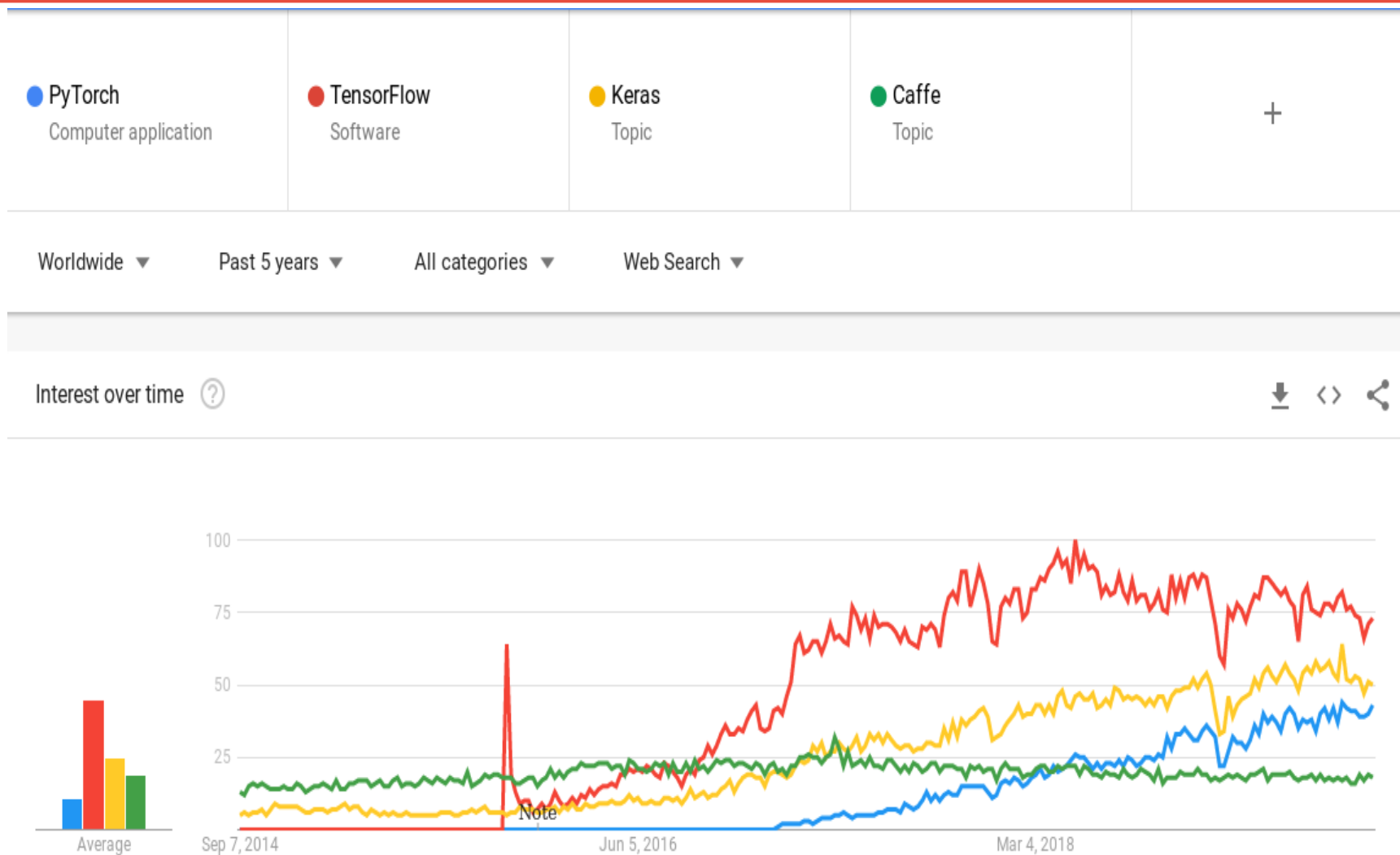


# Comparison (2019)


Deep Learning Framework Six-Month Growth Scores 2019



# Comparison (2014-2019)



# Why Tensorflow?

 tensorflow / tensorflow

Used by ▾48,134

Watch ▾8,587

★ Star133,373

🔗 Fork77,082

<> Code

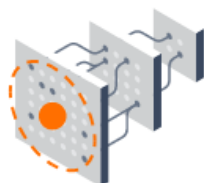
! Issues 2,407

🔗 Pull requests 249

📁 Projects 1

🛡 Security

📊 Insights



## Easy model building

Build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging.



## Robust ML production anywhere

Easily train and deploy models in the cloud, on-prem, in the browser, or on-device no matter what language you use.



## Powerful experimentation for research

A simple and flexible architecture to take new ideas from concept to code, to state-of-the-art models, and to publication faster.

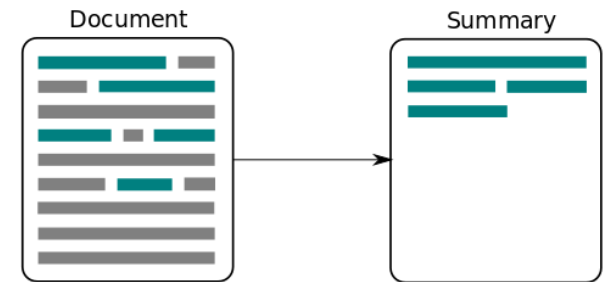
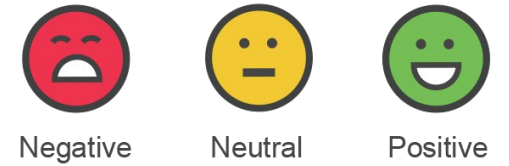
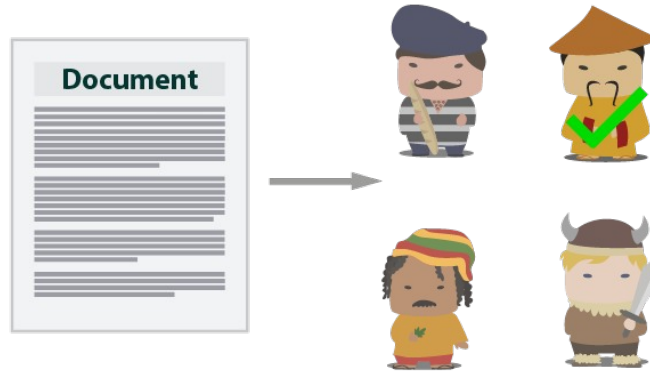
# Features of Tensorflow

- ✓ Open source
  - ✓ Easy Model Building
  - ✓ Popularity
  - ✓ Good support of ML and DL concepts
  - ✓ Multiple CPU/GPU computing
  - ✓ High Scalability
  - ✓ Fast Compile
  - ✓ Documentation
  - ✓ **Tensorboard** for Visualization
  - ✓ Data and model parallelism
  - ✓ Multi-Language Support
  - ✓ Large Community
  - ✓ Production ready
- ✓ Only supports Nvidia GPUs
  - ✓ Static computational graph  
(in comparison with PyTorch)
  - ✓ Debugging  
(in comparison with PyTorch)
  - ✓ Steep Learning Curve  
(in comparison with PyTorch)
  - ✓ Speed  
(in comparison with PyTorch)





# Applications of Tensorflow



# Tensorflow Installation

# How to Install/Use Tensorflow?

## ✓ Install Tensorflow Manually

- Ubuntu 16.04 or later
- macOS 10.12.6 (Sierra) or later (no GPU support)
- Windows 7 or later
- Raspbian 9.0 or later

## ✓ Run a Tensorflow Container

- [Tensorflow Docker Images in Docker Hub](#)

## ✓ Google Colab

# Tensorflow Requirements

## ✓ Hardware requirements for CPU

- Starting with TensorFlow 1.6
- binaries use [AVX instructions](#) which may not run on older CPUs.

## ✓ Hardware requirements for GPU

- NVIDIA® GPU card with CUDA® Compute Capability 3.5 or higher
- See the list of [CUDA-enabled GPU cards](#)

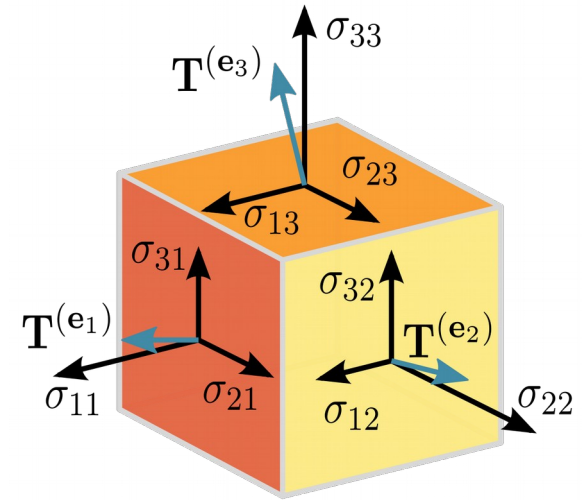
## ✓ Software requirements for GPU

- [NVIDIA® GPU drivers](#) —CUDA 10.0 requires 410.x or higher.
- [CUDA® Toolkit](#) —TensorFlow supports CUDA 10.0 (TensorFlow  $\geq$  1.13.0)
- [CUPTI](#) ships with the CUDA Toolkit.
- [cuDNN SDK](#) ( $\geq$  7.4.1)
- (Optional) [TensorRT](#) 5.0 to improve latency and throughput for inference on some models.

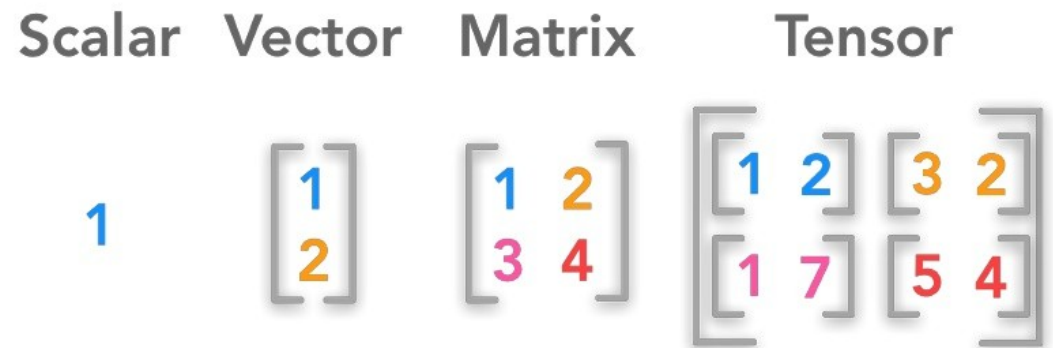
# Tensorflow Basics

# What is a Tensor?

- ✓ The basic data structure in Tensorflow
- ✓ A Mathematic (linear algebra) concept
- ✓ Generalization of vectors and matrices



- ✓ **Typed**
- ✓ **Multidimensional array**
- ✓ **With additional operations**
- ✓ **Modeled in tensor object**



# What is a Tensor?

**Scalar**



**Vector**



**Matrix**



**Tensor**



# Tensor Properties

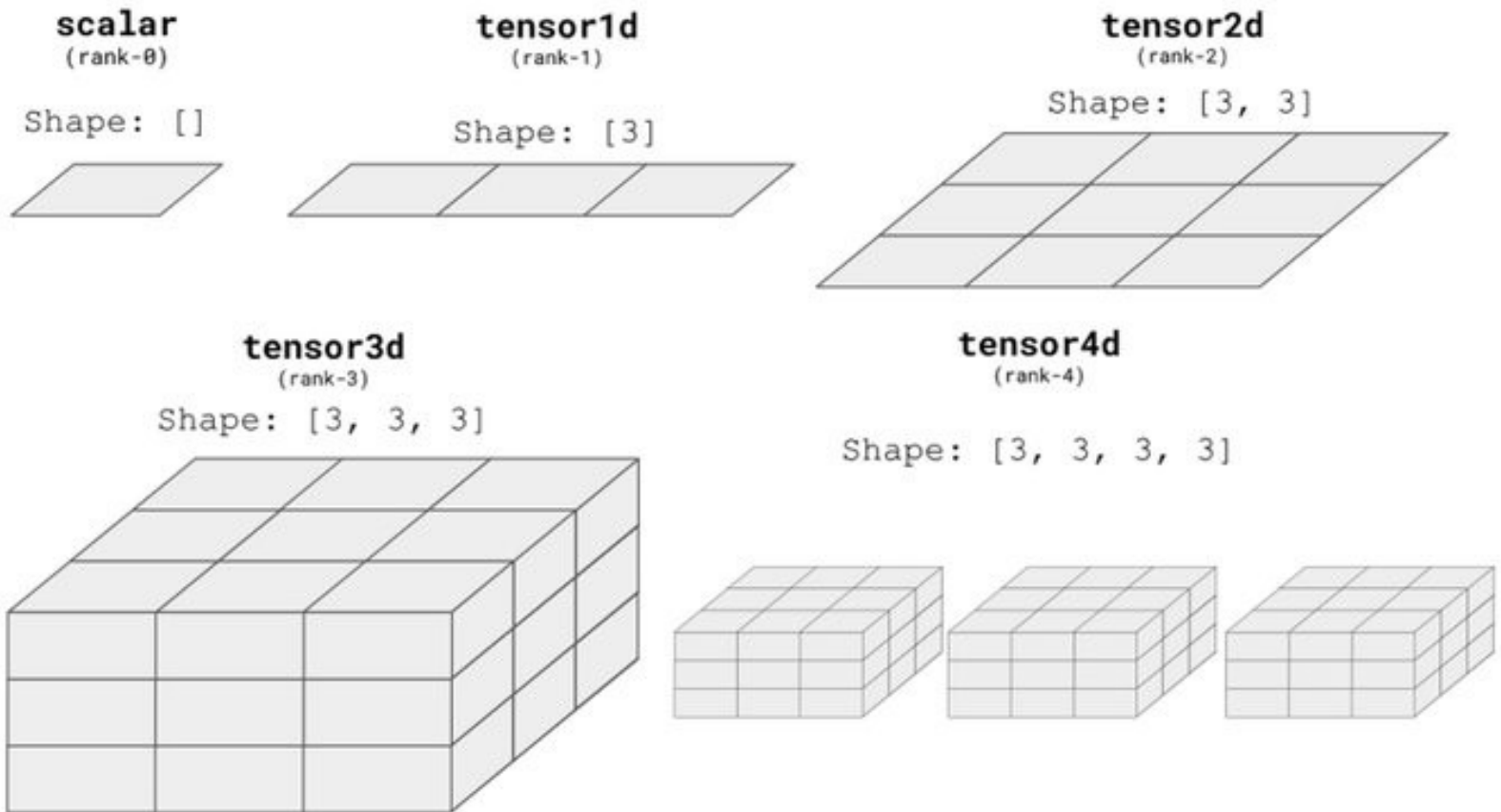
Rank	Entity	Shape	Example		
0	Scalar	[]	4.5	shape=()	dtype=float32
			'hi'		dtype=string
1	Vector	$[D_0]$	[1,2,3]	shape=(3, )	dtype=int32
2	Matrix	$[D_0, D_1]$	[ [1,2,3,4], [5,6,7,8] ]	shape=(2, 4)	dtype=int32
3	3-tensor	$[D_0, D_1, D_2]$	[ [ [1,2], [3,4] ], [ [5,6], [7,8] ] ]	shape=(2, 2, 2)	dtype=int32
n	n-tensor	$[D_0, D_1, ..., D_{n-1}]$	...	...	



# Tensor Properties

- ✓ Rank
- ✓ Shape
- ✓ Type

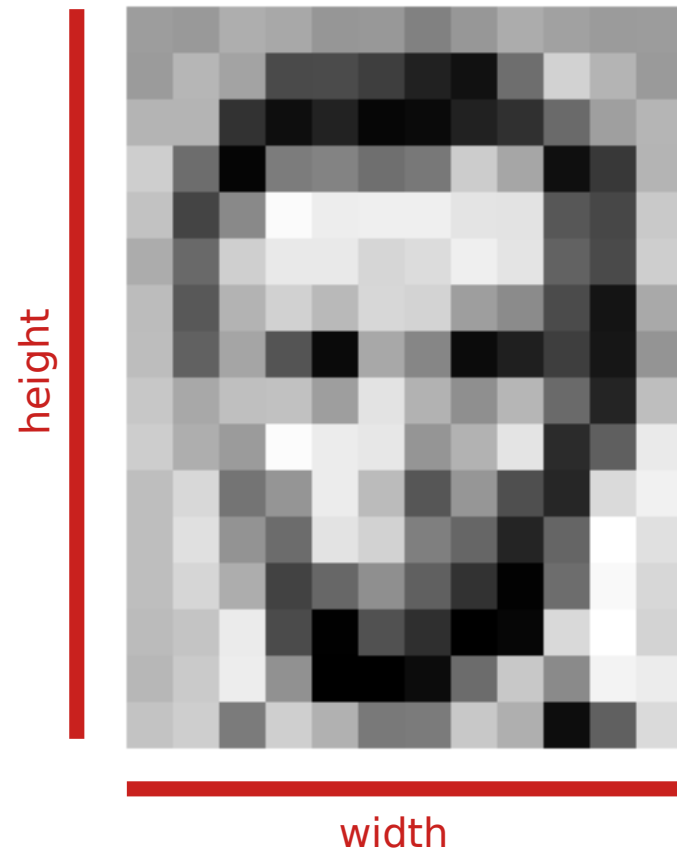
Figure A.1 Examples of tensor of rank 0, 1, 2, 3, and 4.



# Tensor Data Types

Data type	Python type	Description
<code>DT_FLOAT</code>	<code>tf.float32</code>	32 bits floating point.
<code>DT_DOUBLE</code>	<code>tf.float64</code>	64 bits floating point.
<code>DT_INT8</code>	<code>tf.int8</code>	8 bits signed integer.
<code>DT_INT16</code>	<code>tf.int16</code>	16 bits signed integer.
<code>DT_INT32</code>	<code>tf.int32</code>	32 bits signed integer.
<code>DT_INT64</code>	<code>tf.int64</code>	64 bits signed integer.
<code>DT_UINT8</code>	<code>tf.uint8</code>	8 bits unsigned integer.
<code>DT_STRING</code>	<code>tf.string</code>	Variable length byte arrays. Each element of a tensor is a byte array.
<code>DT_BOOL</code>	<code>tf.bool</code>	Boolean.

# Tensors are Everywhere!



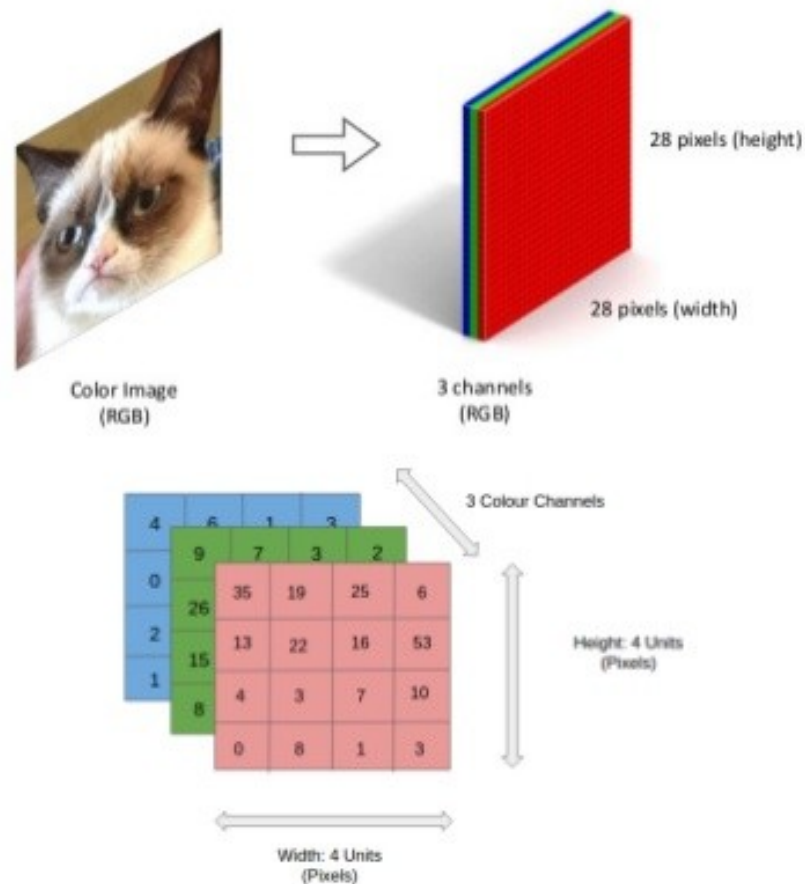
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	165	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

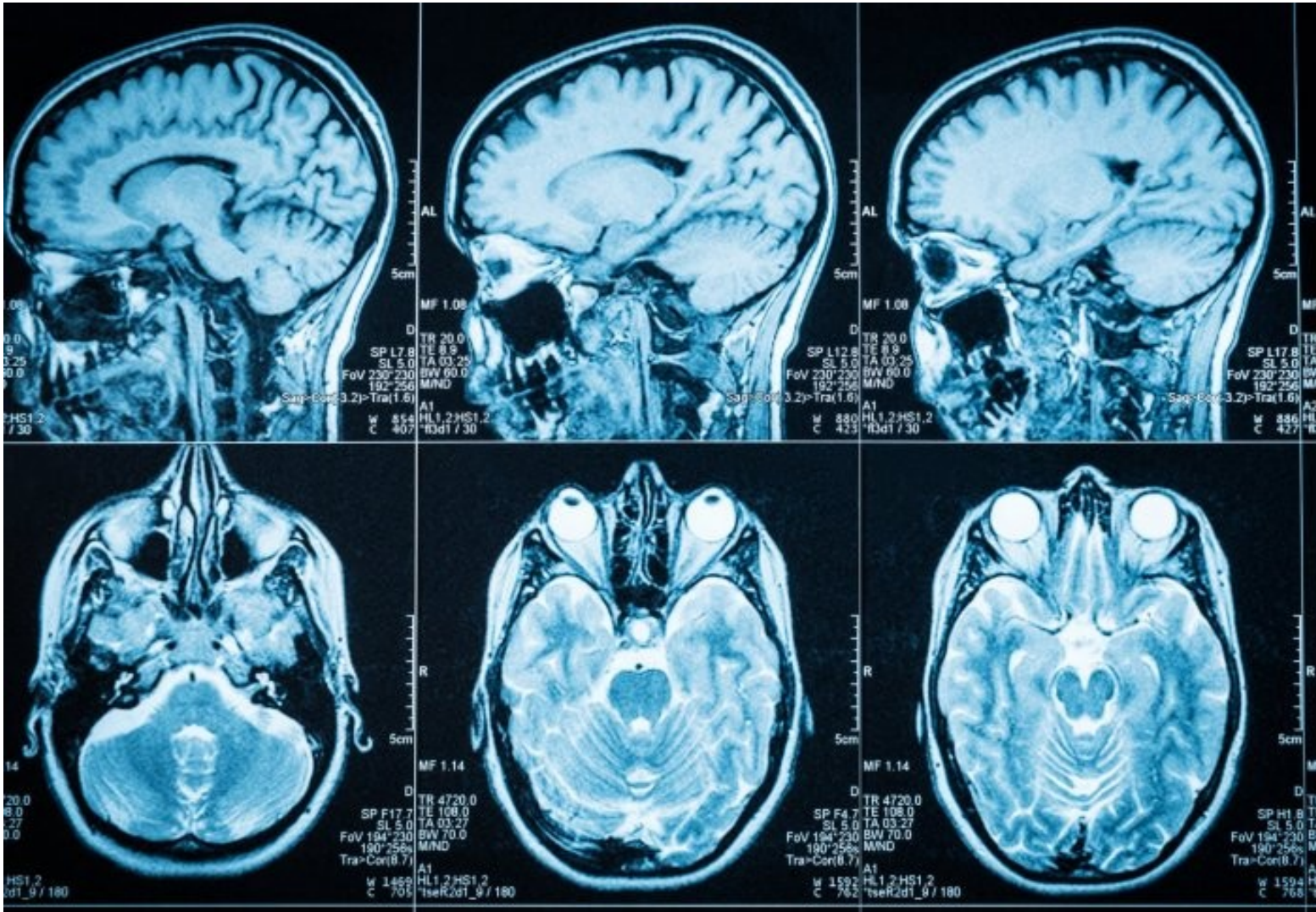
height

# Tensors are Everywhere!

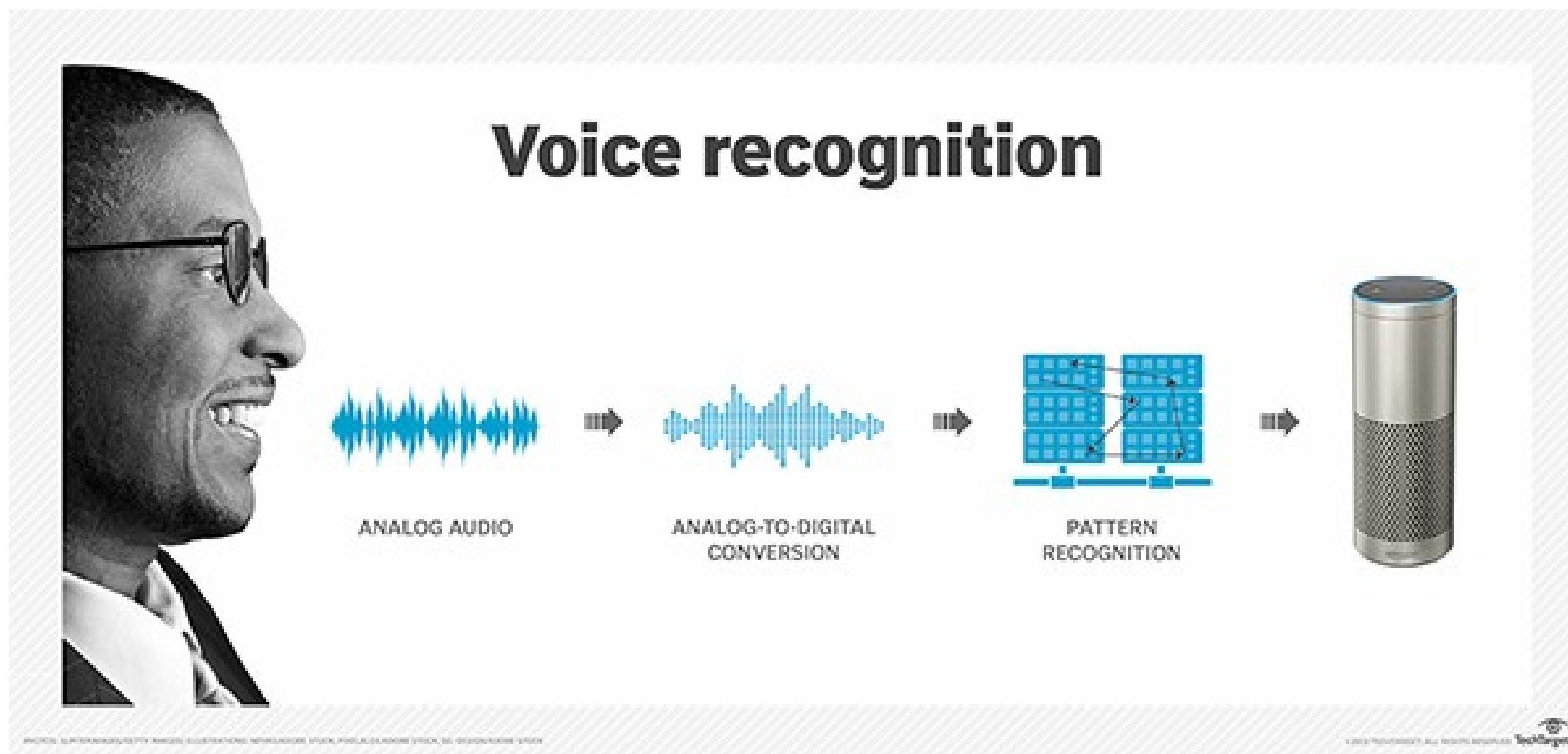
color image is 3rd-order tensor



# Tensors are Everywhere!

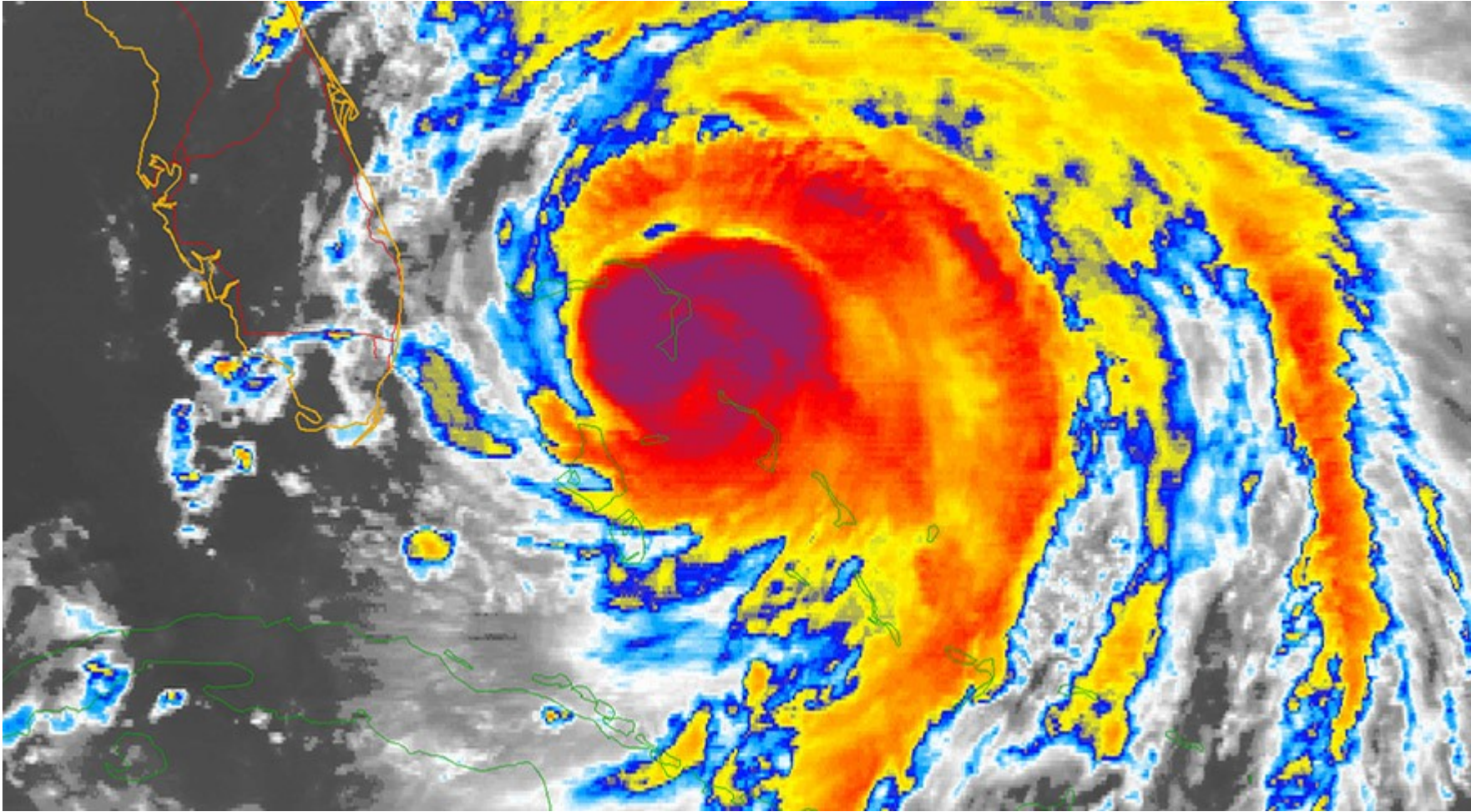


# Tensors are Everywhere!



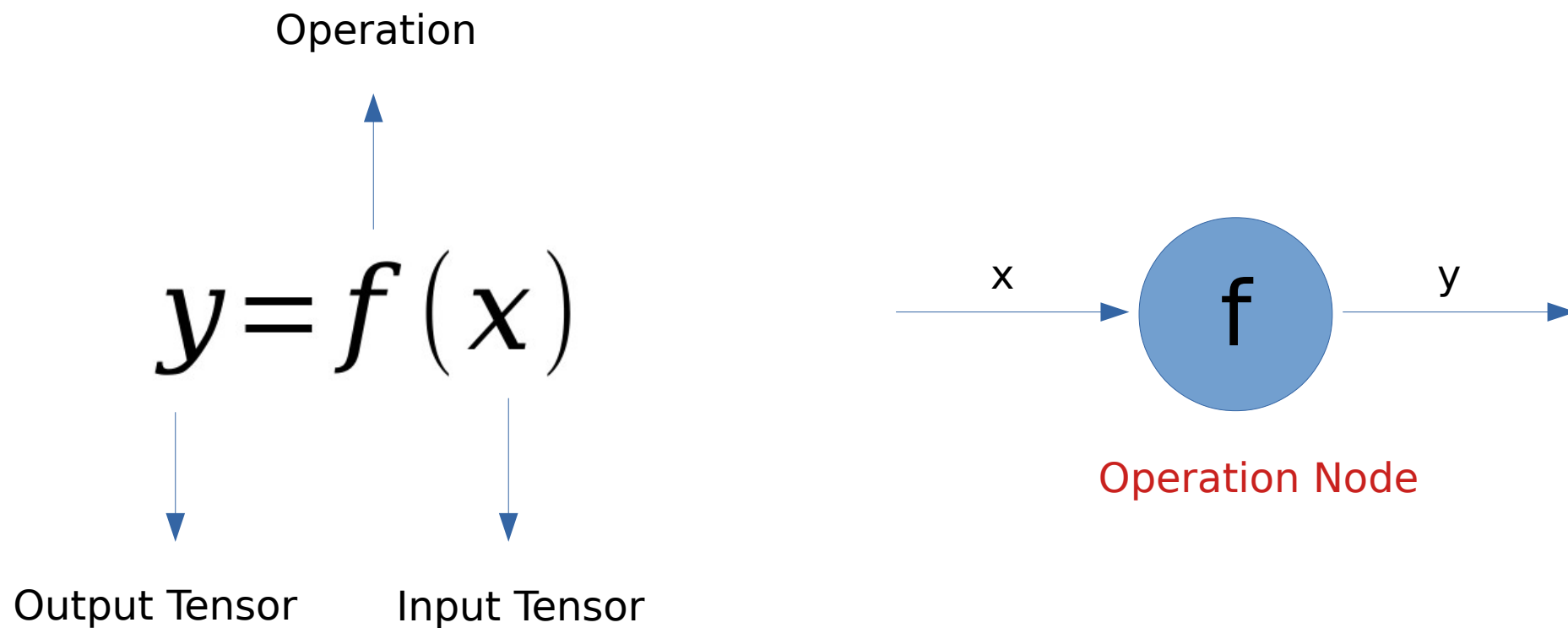


# Tensors are Everywhere!



# Operations or Ops

- ✓ **Anything you can do with a Tensor is an Operation**





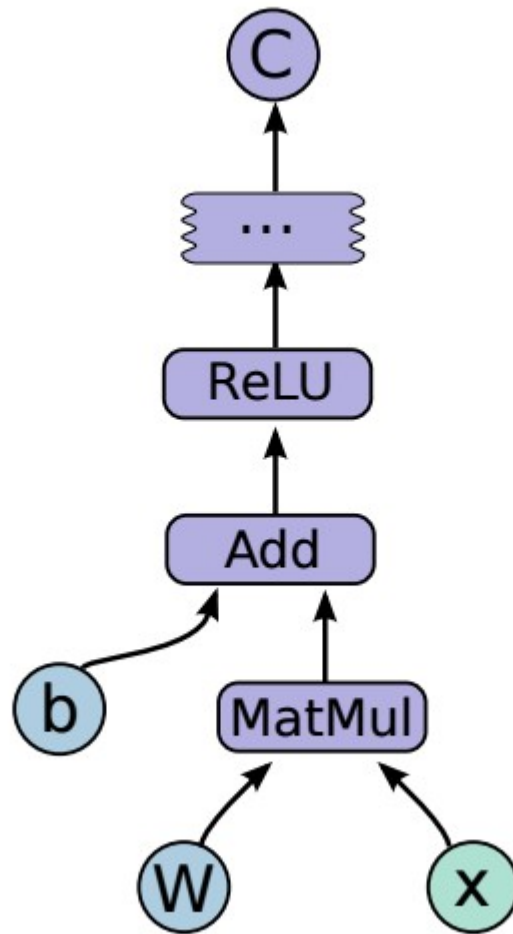
# Computational Graph

- ✓ Composition of **Operation nodes**
- ✓ Every **equation** in **linear algebra** could be demonstrated as **graph**

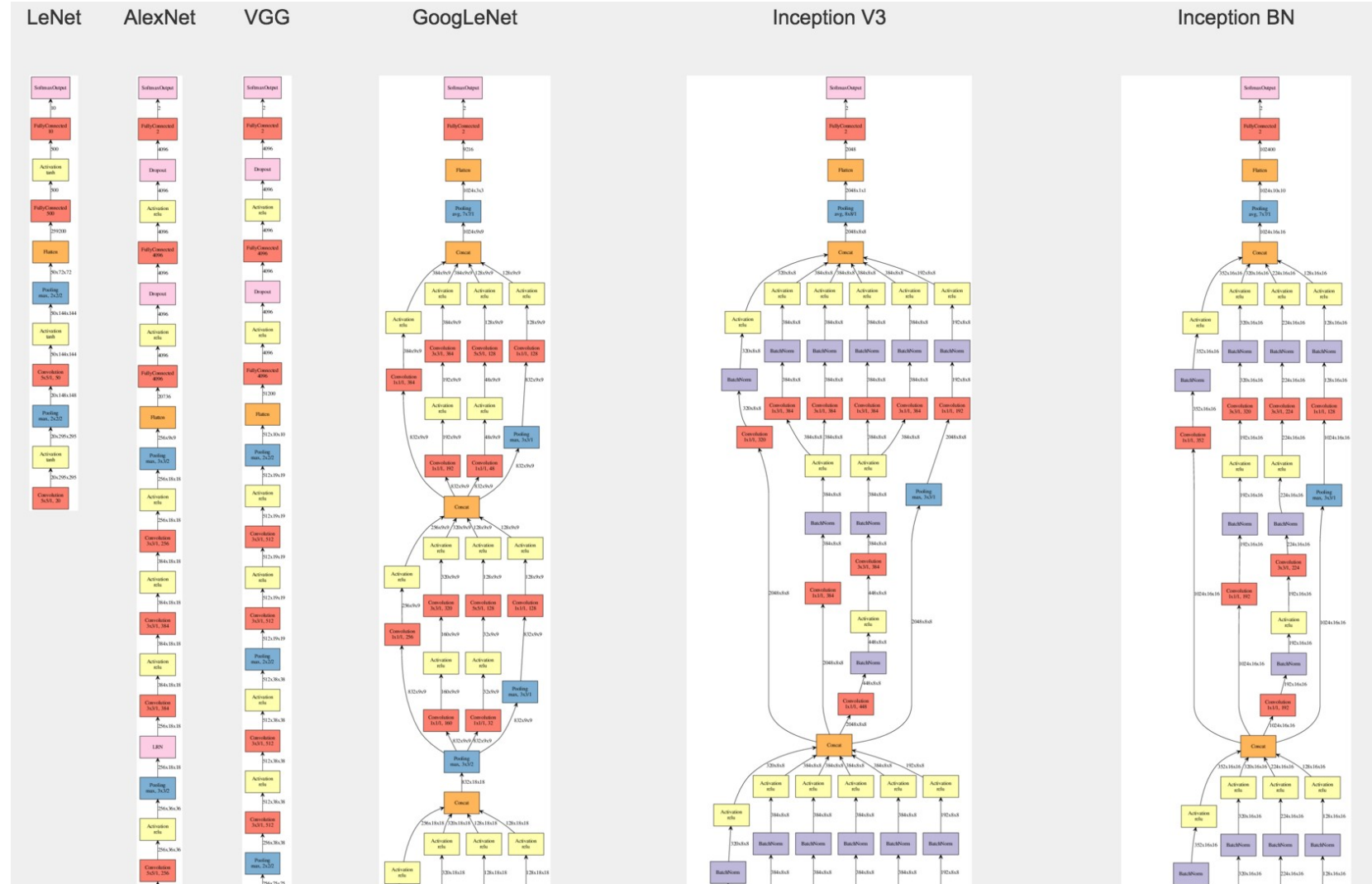
$$y = f(g(x))$$



# Computational Graph



# Computational Graph



# Complicated Computational Graph

## ✓ Needs a powerful framework

- Algorithmic Optimization
- Parallel/Distributed Support
- Memory Efficient
- Multi-Device
- Fast Op Implementation
- Visualization