Introduction t 

Fast Campus
Start Deep Learning with TensorFlow

TensorFlow

A multidimensional array.

A graph of operations.

History of DL Tools

- Mark 1 Perceptron – 1960
- Torch – 2002
- CUDA – 2007
- Theano – 2008
- Caffe – 2014
- DistBelief – 2011
- TensorFlow 0.1 – 2015
- PyTorch 0.1 – 2017
- TensorFlow 1.0 – 2017
- PyTorch 1.0 – 2017
- TensorFlow 2.0 – 2019

TensorFlow in One Slide

- **What is it:** Deep Learning Library *(and more)*
  - **Facts:** Open Source, Python, Google
- **Community:**
  - 117,000+ GitHub stars
  - TensorFlow.org: Blogs, Documentation, DevSummit, YouTube talks
- **Ecosystem:**
  - Keras: high-level API
  - TensorFlow.js: in the browser
  - TensorFlow Lite: on the phone
  - Colaboratory: in the cloud
  - TPU: optimized hardware
  - TensorBoard: visualization
  - TensorFlow Hub: graph modules
- **Alternatives**: PyTorch, MXNet, CNTK

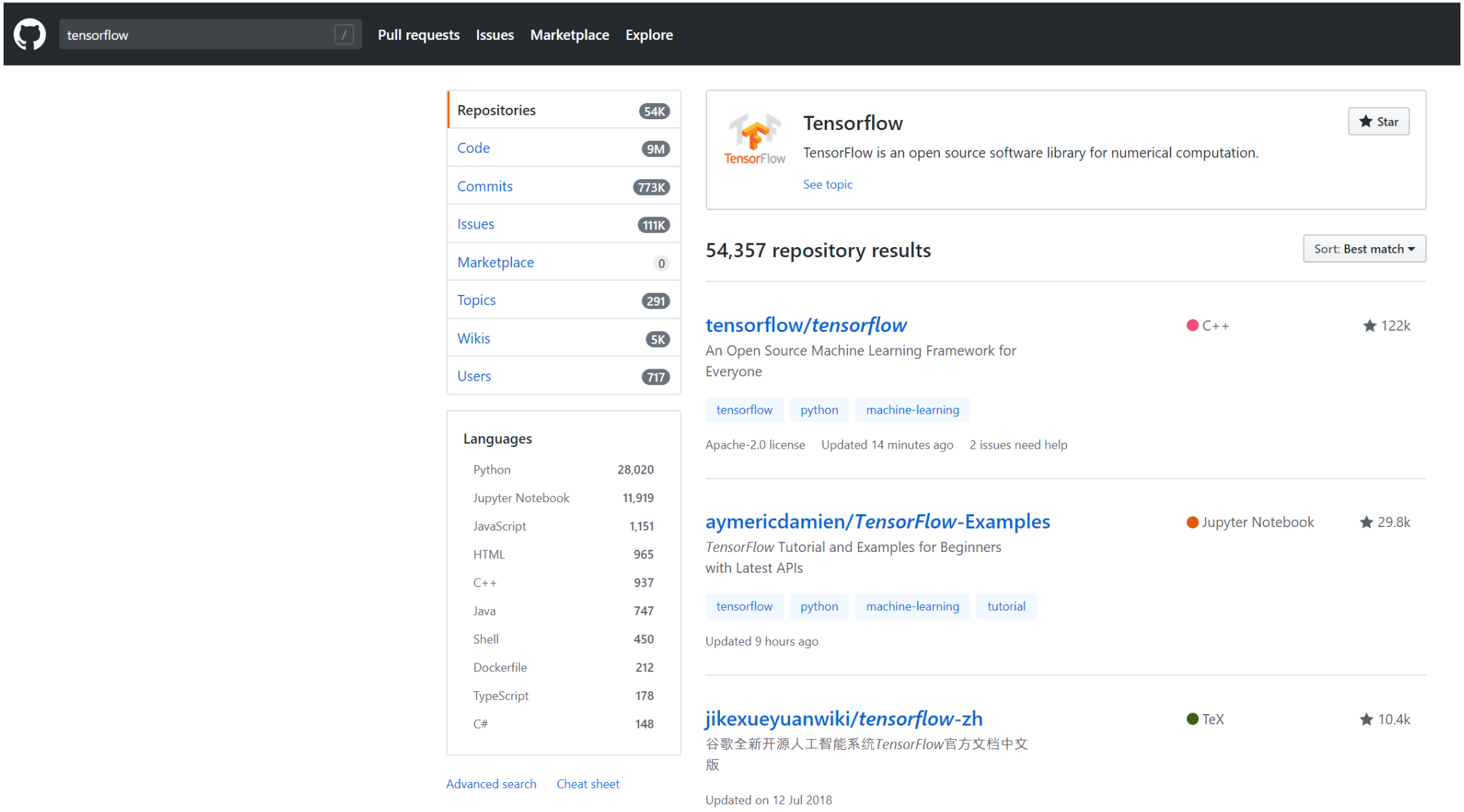
Extras:

- Swift for TensorFlow
- TensorFlow Serving
- TensorFlow Extended (TFX)
- TensorFlow Probability
- Tensor2Tensor

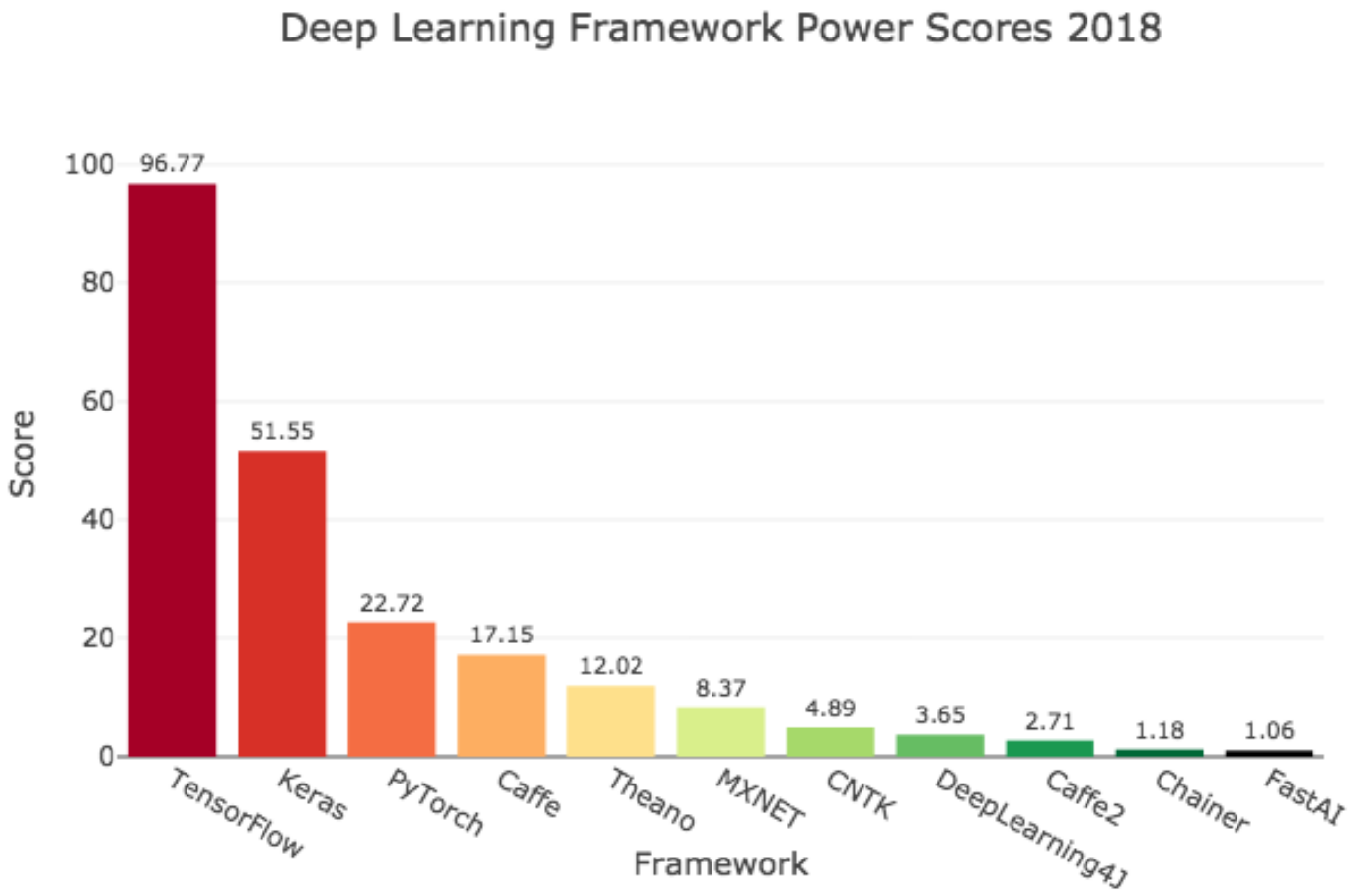
TensorFlow Statistics

41,000,000	50,000+	9,900+	1,800+
Downloads	commits	pull requests	contributors

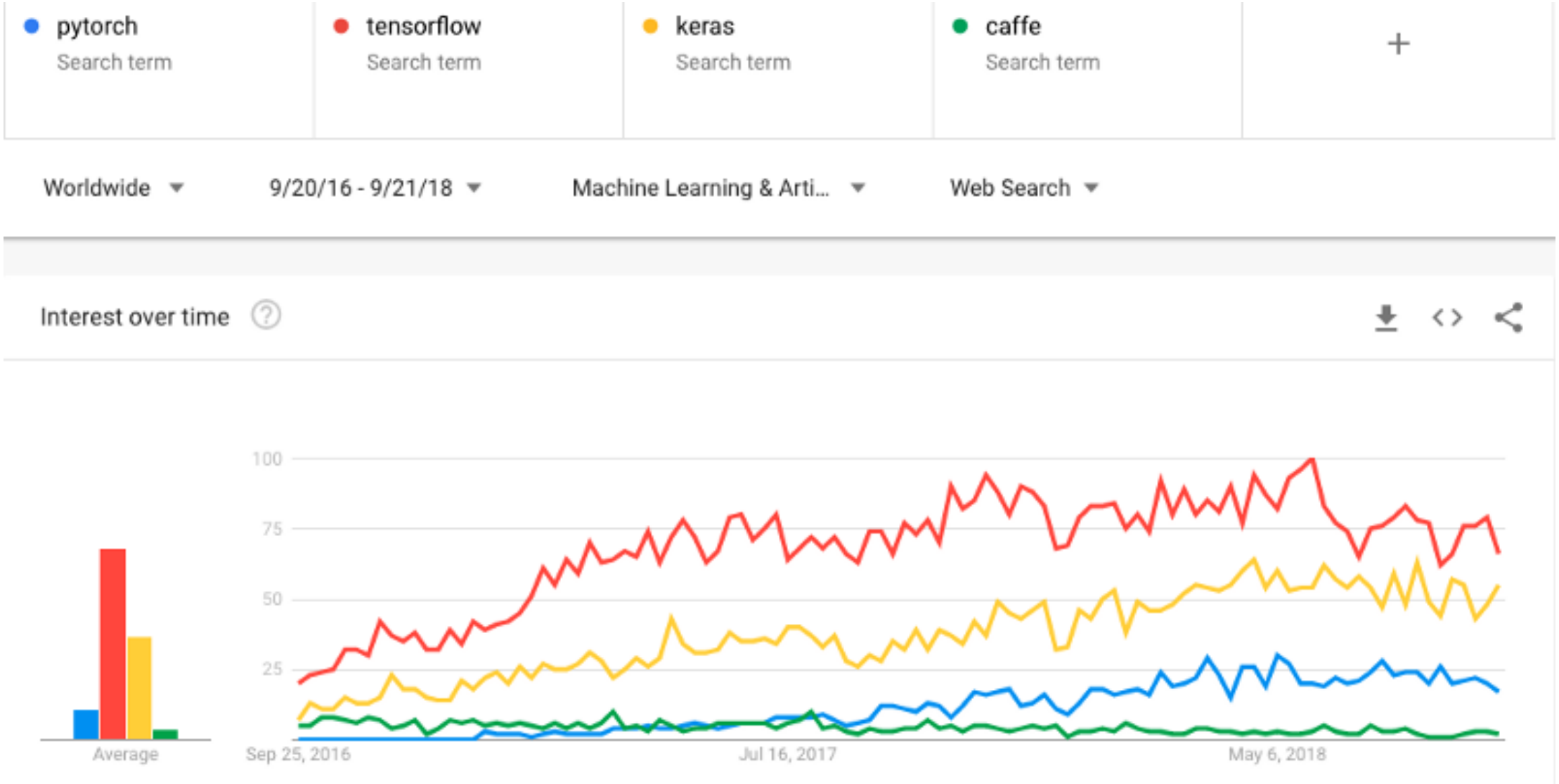
Search Results on Github



Popularity

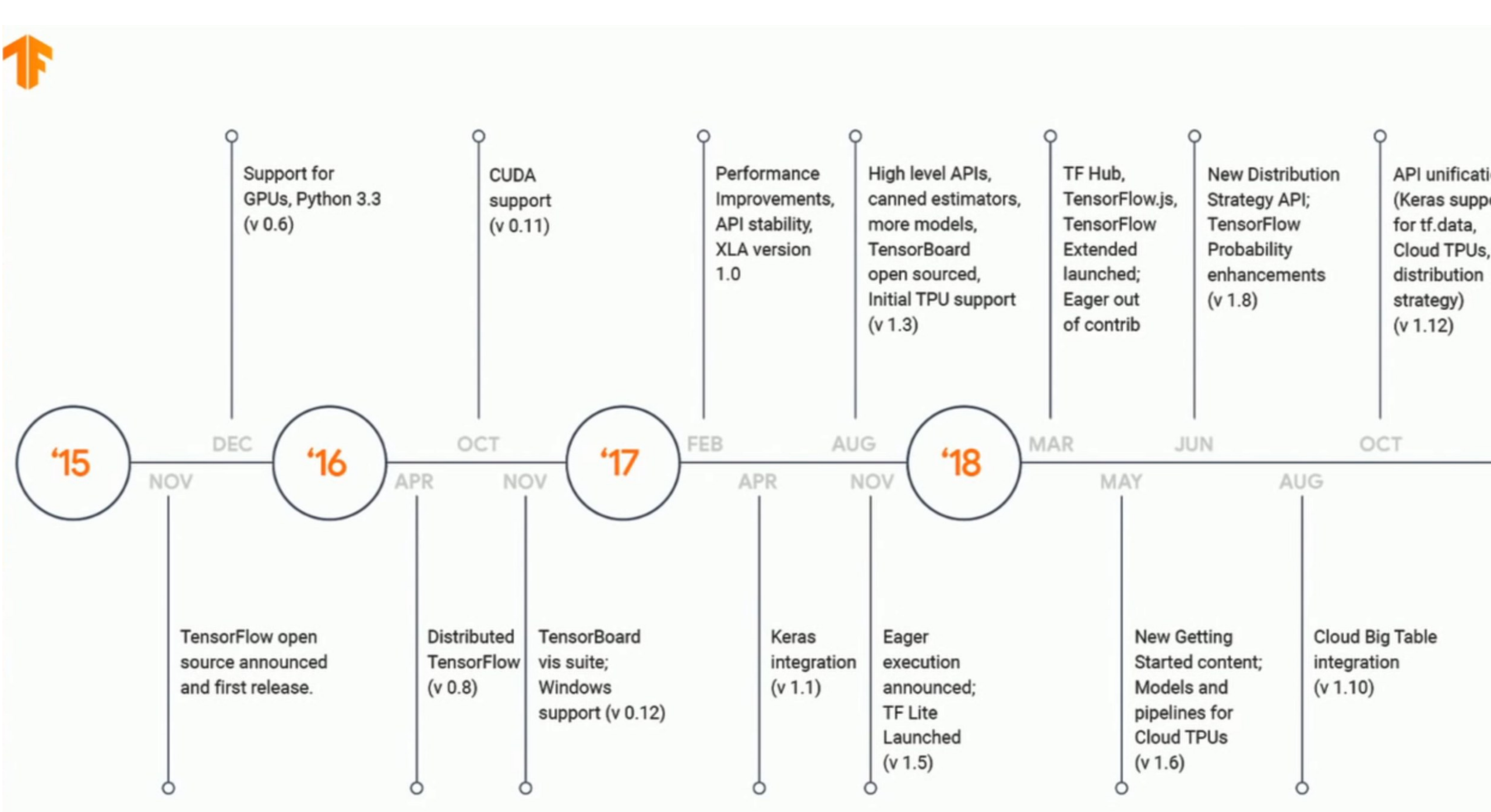

Deep Learning Framework Power Scores 2018
Score
100
80
60
40
20
0
96.77
51.55
22.72
17.15
12.02
8.37
4.89
3.65
2.71
1.18
1.06
TensorFlow
Keras
PyTorch
Caffe
Theano
MXNET
CNTK
DeepLearning4J
Caffe2
Chainer
FastAI
Framework

Google Search Statistics



Companies Using Tensorflow

- Google
- OpenAI
- DeepMind
- Snapchat
- Uber
- Airbus
- eBay
- Dropbox
- A bunch of startups

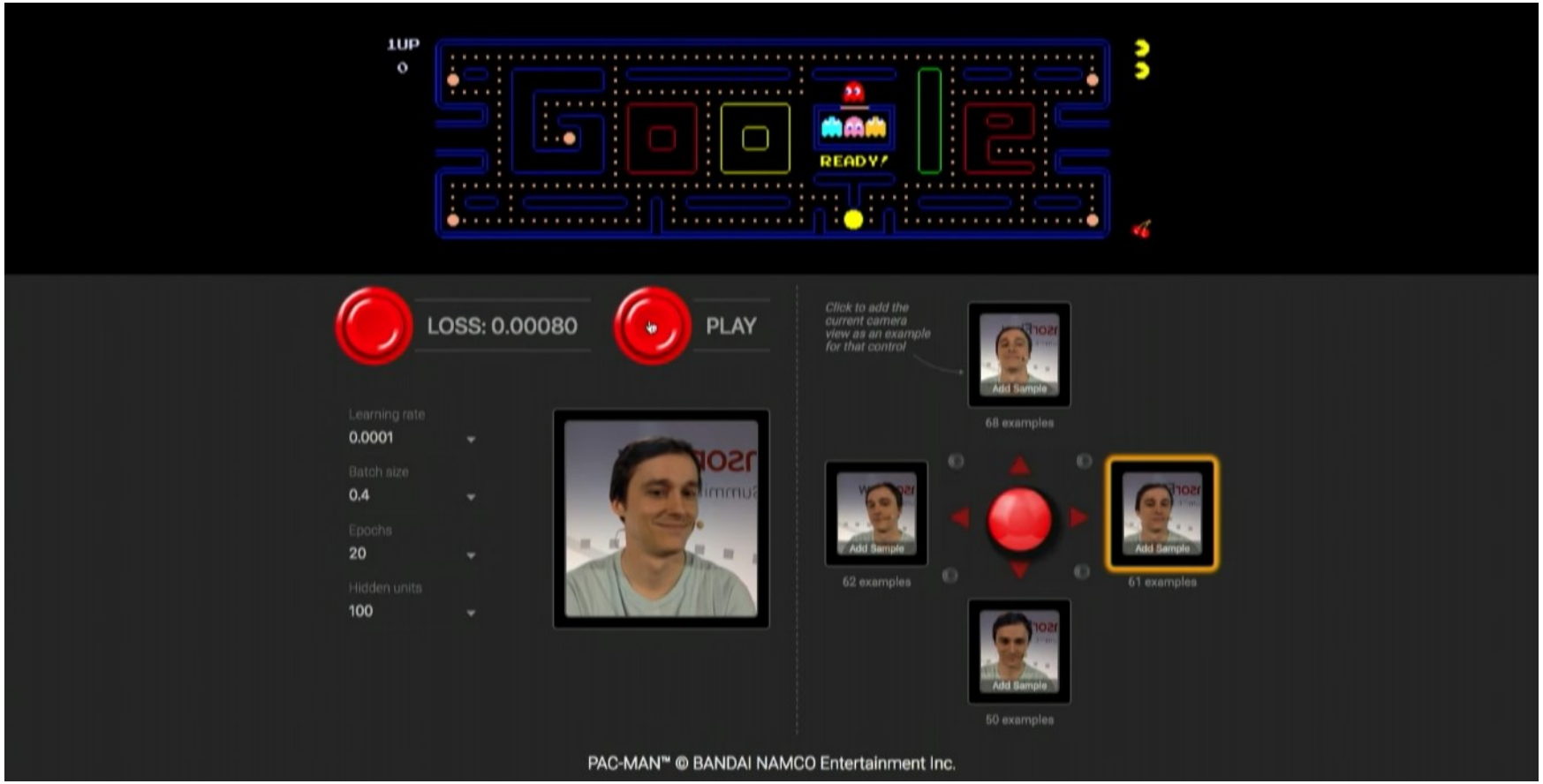

'15
NOV
TensorFlow open source announced and first release.
DEC
Support for GPUs, Python 3.3 (v 0.6)
'16
APR
Distributed TensorFlow (v 0.8)
OCT
CUDA support (v 0.11)
NOV
TensorBoard vis suite; Windows support (v 0.12)
'17
FEB
Performance Improvements, API stability, XLA version 1.0
APR
Keras integration (v 1.1)
AUG
High level APIs, canned estimators, more models, TensorBoard open sourced, Initial TPU support (v 1.3)
NOV
Eager execution announced; TF Lite Launched (v 1.5)
'18
MAR
TF Hub, TensorFlow.js, TensorFlow Extended launched; Eager out of contrib
MAY
New Getting Started content; Models and pipelines for Cloud TPUs (v 1.6)
JUN
New Distribution Strategy API; TensorFlow Probability enhancements (v 1.8)
AUG
Cloud Big Table integration (v 1.10)
OCT
API unificati
(Keras suppo
for tf.data,
Cloud TPUs,
distribution
strategy)
(v 1.12)

TensorFlow.js

- http://bit.ly/pose-net
- http://bit.ly/body-fix

- Magenta (Google)
  - Use machine learning to create compelling art and music. Their projects are really fun! For example, please check out Draw Together with a Neural Network.

- [Webcam controller PacMan]



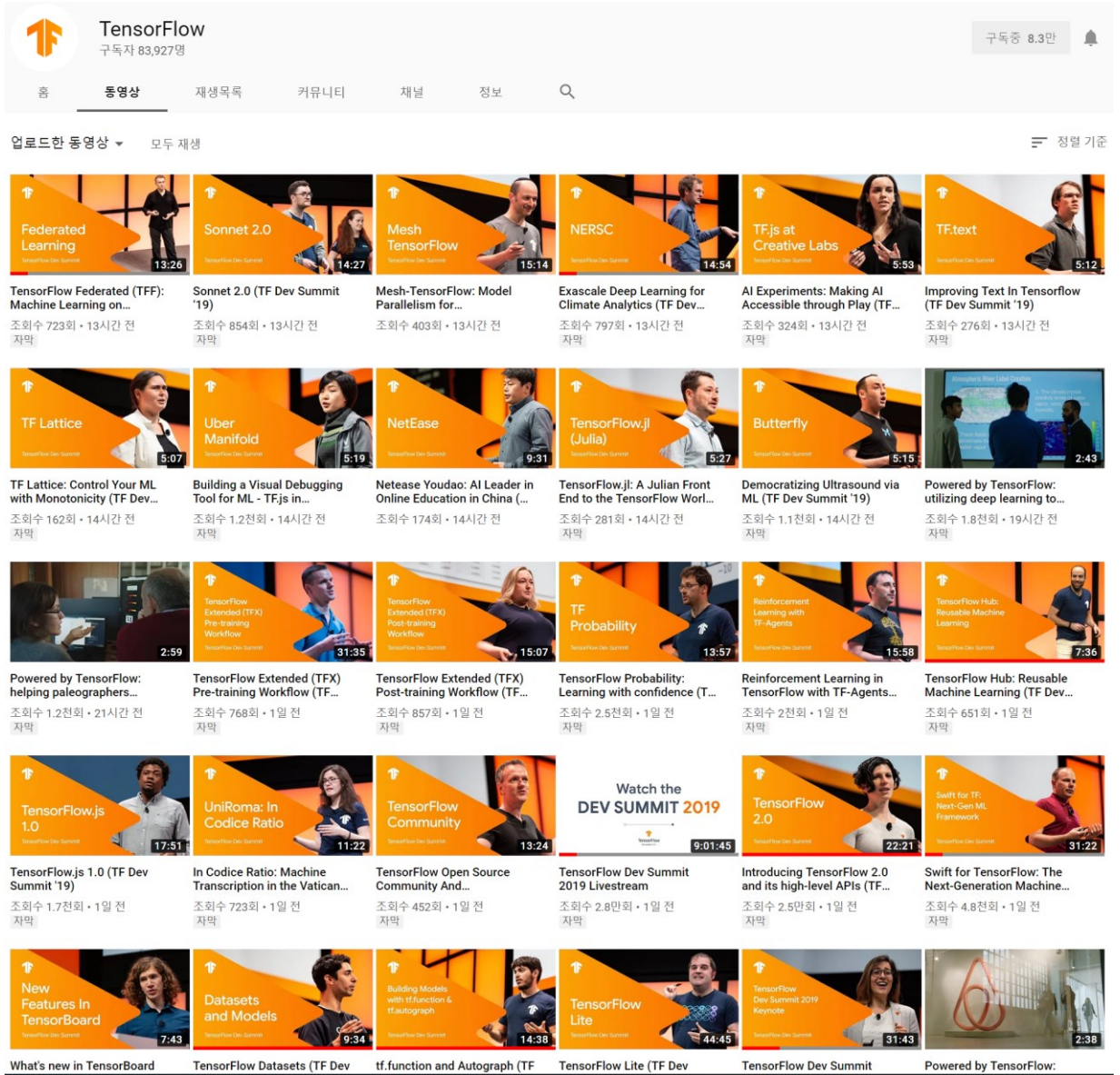
TensorFlow Dev Summit 2019

Recap of the 2019 TensorFlow Dev Summit

TensorFlow
Mar 8 · 7 min read

Posted by Fred Alcober and Sandeep Gupta, on behalf of the TensorFlow team

TensorFlow held its third and biggest yet annual Developer Summit in Sunnyvale, CA on March 6 and 7, 2019. The event was attended by approximately 1,000 machine learning enthusiasts and watched over livestream by tens of thousands more.



TensorFlow 2.0

- 2018년 8월 15일 로드맵 공개 / 2019년 1월 프리뷰 공개
  - 개발 / 사용 양측에서의 사용성 개선
  - 일관성 강화
  - 미래의 최적화를 위한 룸 확보
- API 대대적 정리
  - 다양한 코드를 받아들이며 동일한 기능을 하는 여러 API가 존재함
  - 하나씩만 남기고 모두 제거
- 개발 편의성 개선
  - 디버그 편의성 개선
- Name space의 일관성 유지
  - 전역 변수 형태의 참조 모두 제거
- 대규모 훈련의 편의성 개선
  - Distributed TensorFlow로 별도 관리되던 부분 통합

TensorFlow 2.0

- API cleanup
- tf.keras
- Eager execution
- Functions, not sessions

```
# TensorFlow 1.X
outputs = session.run(f(placeholder), feed_dict={placeholder: input})
# TensorFlow 2.0
outputs = f(input)
```

API Cleanup

- tf.contrib → TF Addons (https://github.com/tensorflow/addons)
- tf.app, tf.flags, tf.logging is removed
- No more Globals: tf.global_variables_initializer(), tf.variable_scope(), etc
- Duplicated API cleanup
- Parameter name and order compatible with numpy (e.g dim → axis)
- Support legacy API at tf.compat.v1

TensorFlow 1.x

```
>>> import tensorflow as tf
>>>
>>> t = tf.nn.sigmoid([0.])
>>>
>>> print(t)

Tensor("Sigmoid_1:0", shape=(1,), dtype=float32)
```

TensorFlow 2.x

```
>>> import tensorflow as tf
>>>
>>> t = tf.nn.sigmoid([0.])
>>>
>>> print(t)

tf.Tensor([0.5], shape=(1,), dtype=float32)

>>>
>>> print(t.numpy())


[0.5]
```

TensorFlow 1.x

```
import tensorflow as tf

## 그래프를 정의합니다
g = tf.Graph()
with g.as_default():
    x = tf.placeholder(dtype=tf.float32,
                       shape=(None), name='x')
    w = tf.Variable(2.0, name='weight')
    b = tf.Variable(0.7, name='bias')
    z = w * x + b
    init = tf.global_variables_initializer()

## 세션을 만들고 그래프 g를 전달합니다
with tf.Session(graph=g) as sess:
    ## w와 b를 초기화합니다
    sess.run(init)
    ## z를 평가합니다
    for t in [1.0, 0.6, -1.8]:
        print('x=%4.1f --> z=%4.1f'%(
              t, sess.run(z, feed_dict={x:t})))
```

TensorFlow 2.x

```
import tensorflow as tf

w = tf.Variable(2.0, name='weight')
b = tf.Variable(0.7, name='bias')

# z를 평가합니다
for x in [1.0, 0.6, -1.8]:
    z = w * x + b
    print('x=%4.1f --> z=%4.1f'%(x, z))
```