



## 08 TensorFlow2.0基础

西安科技大学 牟琦  
muqi@xust.edu.cn



## 8.1 TensorFlow2.0特性

# An end-to-end open source machine learning platform

□ **end-to-end**: 端到端

□ **open source**: 开放源代码, 开放设计和实现框架

*TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*

*Wide & Deep Learning for Recommender Systems*

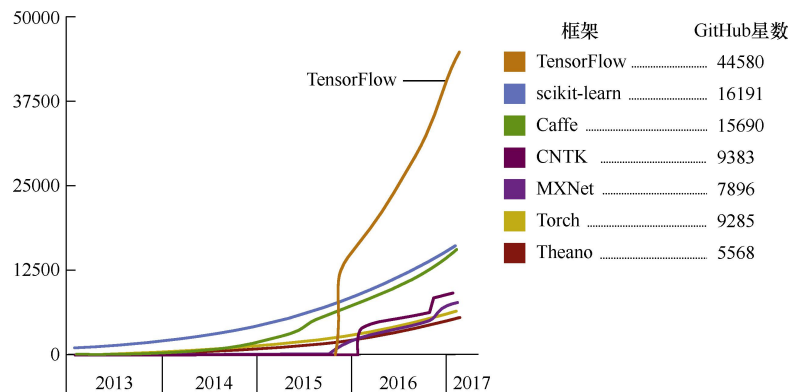
*The YouTube Video Recommendation System*

□ **machine learning**: 机器学习生态系统

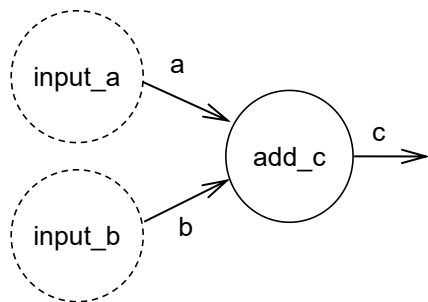


### TensorFlow的发展历程

- 2011: DistBelief
- 2015.11: TensorFlow 0.5.0
- 2017.02: TensorFlow 1.0
  - 高层API, 将Keras库整合进其中
  - 动态图机制: Eager Execution
  - 面向移动智能终端: TensorFlow Lite
  - 面向网页前端: TensorFlow.js
  - 自动生成计算图: AutoGraph
- 2019: TensorFlow 2.0



### TensorFlow1.x——延迟执行机制 (deferred execution) / 静态图机制



```
a=tf.constant(2,name="input_a")  
b=tf.constant(3,name="input_b")  
c=tf.add(a,b,name="add_c")
```

```
sess=tf.Session()  
print(sess.run(c))  
sess.close()
```

构建阶段

只是描述了计算图，  
并没有实质的计算发生

执行阶段

在会话中运行计算图

- 代码运行效率高，便于优化
- 程序不够简洁



### ■ TensorFlow2.0——动态图机制 (Eager Execution)

```
a=tf.constant(2,name="input_a")  
b=tf.constant(3,name="input_b")  
print(a+b)
```

- 无需首先创建静态图，可以立刻执行计算，并返回结果
- 能够快速的建立和调试模型
- 执行效率不高

兼顾**易用性**和**执行效率**——

在**程序调试**阶段使用**动态图**，快速建立模型、调试程序；

在**部署阶段**，采用**静态图**机制，从而提高模型的性能和部署能力



### ■ TensorFlow1.x——重复、冗余的API

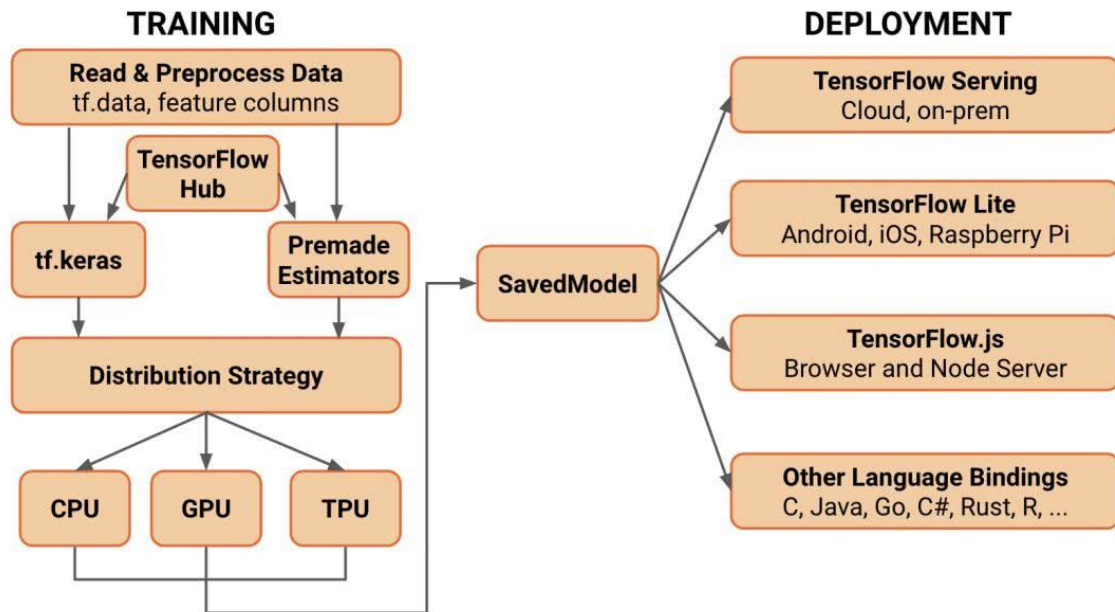
构建神经网络: `tf.slim`, `tf.layers`, `tf.contrib.layers`, `tf.keras`  
混乱, 不利于程序共享, 维护的成本高

### ■ TensorFlow2.0——清理 / 整合API

清理、整合了重复的API  
将`tf.keras`作为构建和训练模型的标准高级API



### TensorFlow2.0架构





### ■ TensorFlow框架特性

#### □ 多种环境支持

可运行于移动设备、个人计算机、服务器、集群等  
云端、本地、浏览器、移动设备、嵌入式设备

#### □ 支持分布式模式

TensorFlow会自动检测GPU和CPU，并充分利用它们并行、分布的执行

#### □ 简洁高效

构建、训练、迭代模型：Eager Execution, Keras  
部署阶段：转化为静态图，提高执行效率。

#### □ 社区支持



## 8.1 TensorFlow2.0特性

### 使用 TensorFlow提供解决方案的部分企业

