

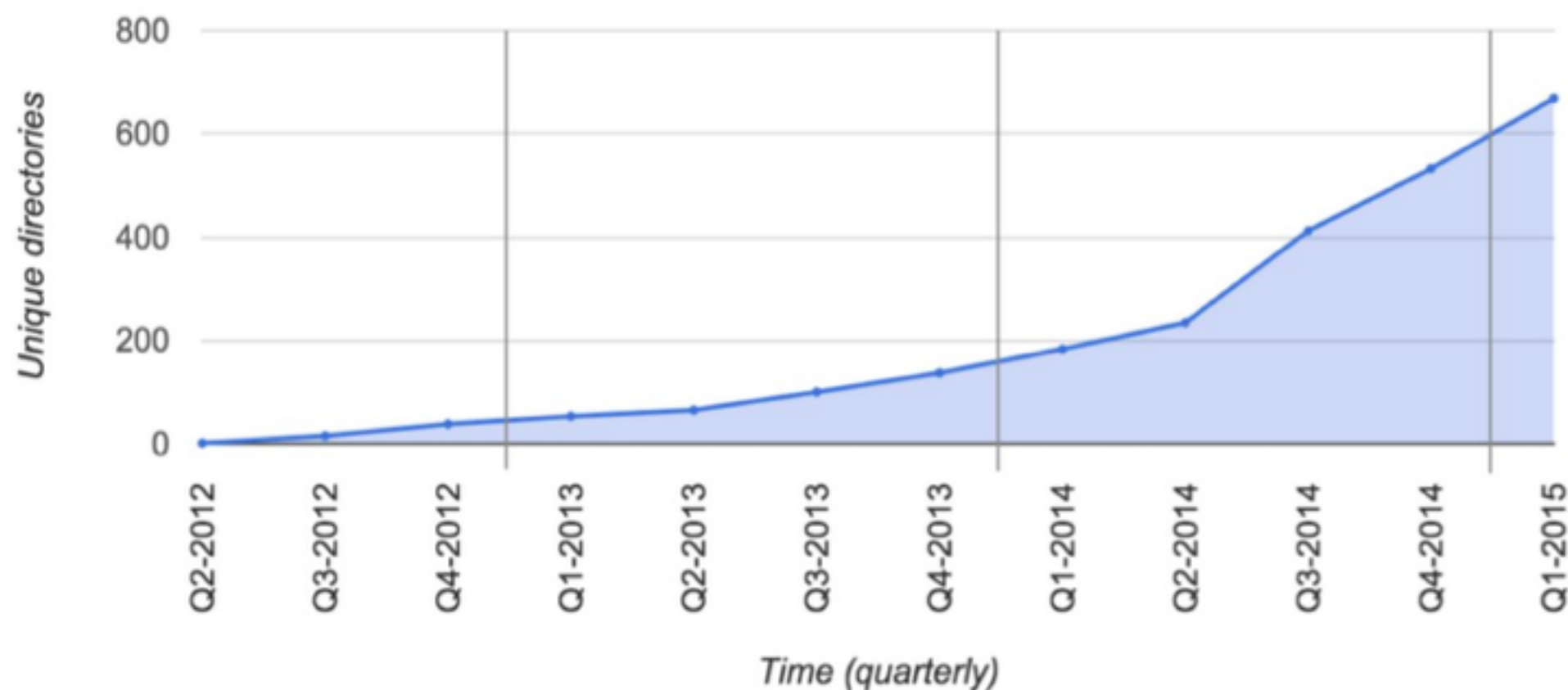


# TensorFlow

一个异构分布式系统的大规模机器学习系统

# 深度学习在Google的应用

Android  
Apps  
Gmail  
Image Understanding  
Maps  
NLP  
Photos  
Robotics  
Speech  
Translation  
many research uses..  
YouTube  
... many others



# 神经网络的重要特性

**Results get better with**

- **more data +**
- **bigger models +**
- **more computation**

# 迭代周期与效果

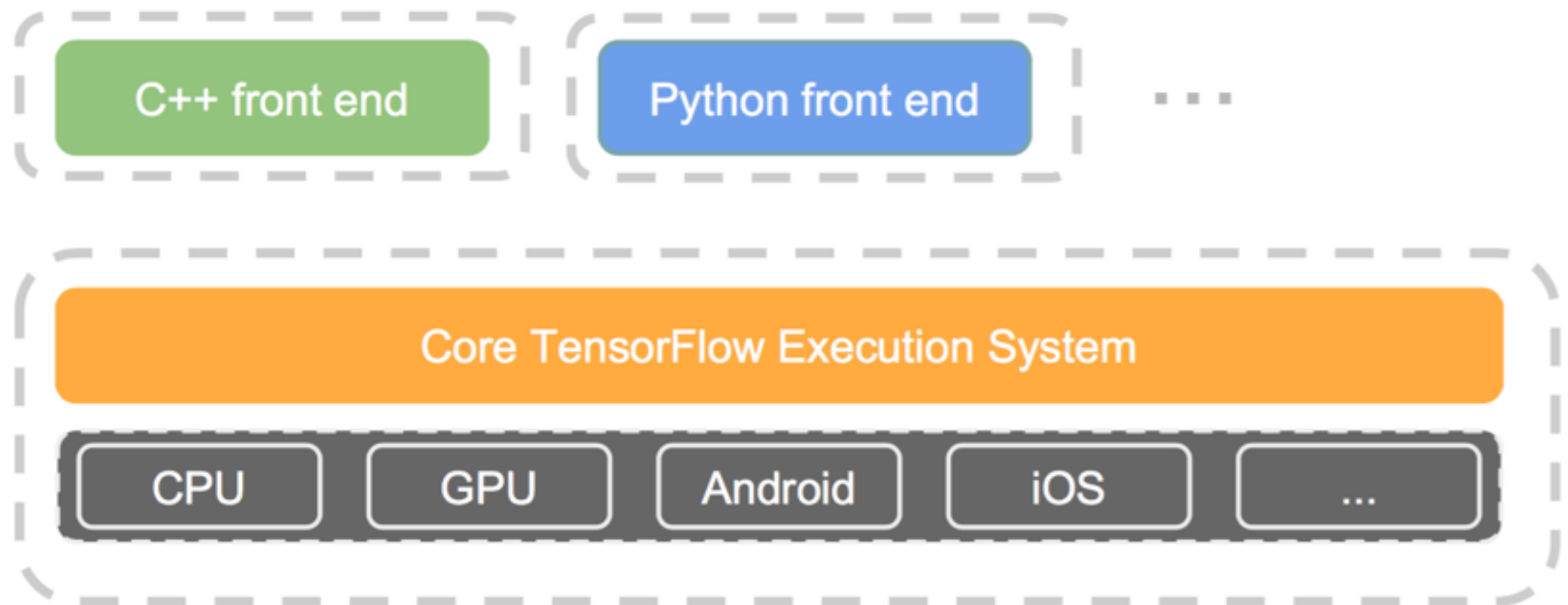
- Minutes, Hours:
  - **Interactive research! Instant gratification!**
- 1-4 days
  - Tolerable
  - Interactivity replaced by running many experiments in parallel
- 1-4 weeks:
  - High value experiments only
  - Progress stalls
- >1 month
  - Don't even try

# TensorFlow特点

- Google第二代深度学习系统
- 目前支持CNN、RNN和LSTM算法
- 跨平台运行
  - 移动设备、单机、集群
- 弹性 (flexible)
- 可扩展 (Extensible)

# TensorFlow特点

- 核心部分C++实现
- 支持不同的前端
  - 当前支持python、C++
- 可扩展



# 安装

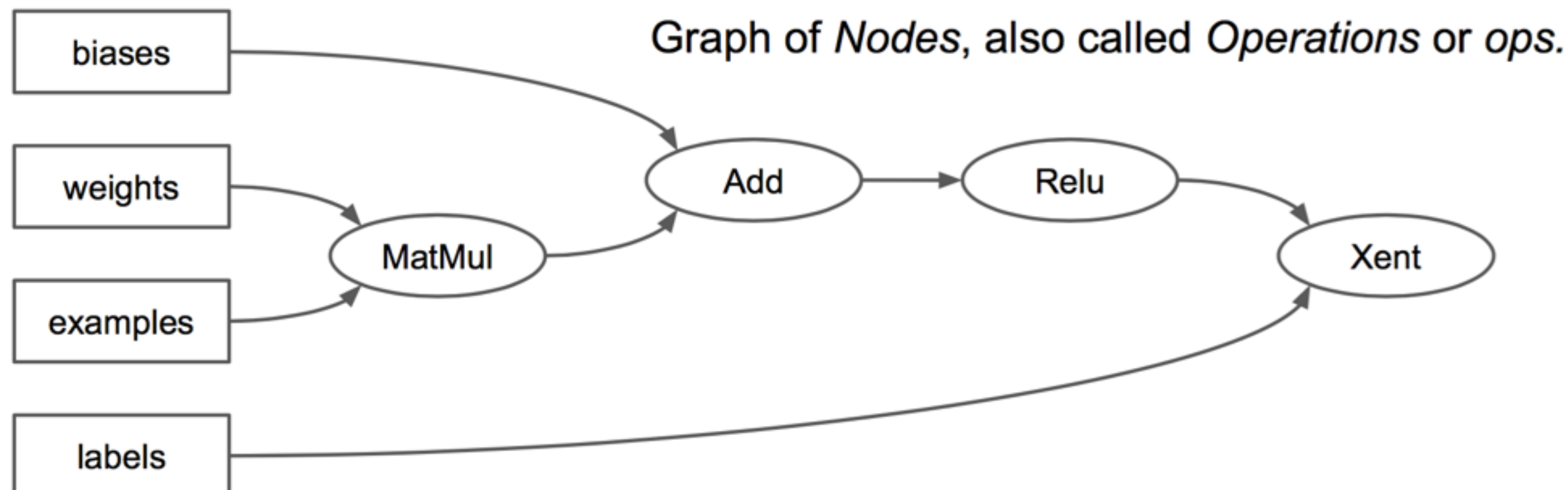
- 源码: <https://github.com/tensorflow/tensorflow>
  - 当前仅开源单机版
- 安装命令
  - Ubuntu/Linux
    - # 仅使用 CPU 的版本
    - \$ pip install [https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-0.5.0-cp27-none-linux\\_x86\\_64.whl](https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-0.5.0-cp27-none-linux_x86_64.whl)
    - # 开启 GPU 支持的版本 (安装该版本的前提是已经安装了 CUDA sdk)
    - \$ pip install [https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow-0.5.0-cp27-none-linux\\_x86\\_64.whl](https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow-0.5.0-cp27-none-linux_x86_64.whl)
  - Mac OS X
    - # 当前版本只支持 CPU
    - \$ pip install <https://storage.googleapis.com/tensorflow/mac/tensorflow-0.5.0-py2-none-any.whl>

# 编程模型

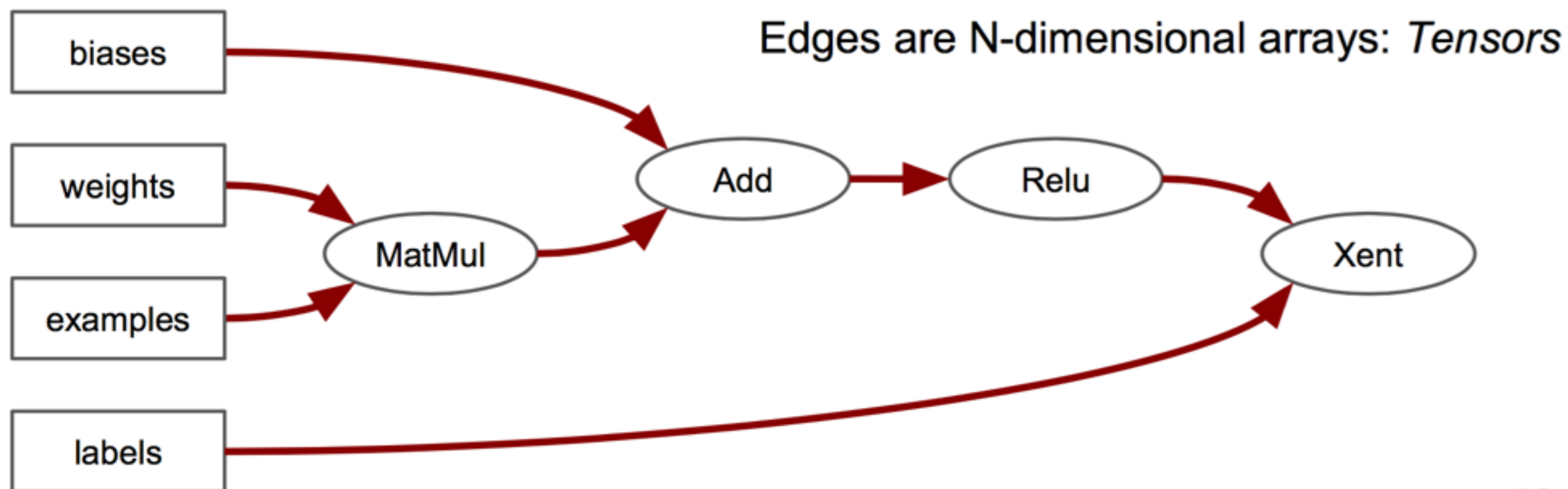
- 有向图描述计算
  - 节点集合组成
  - 一些类型的节点保持和更新持久状态
  - 分支和循环结构
- 在 TensorFlow 图中
  - 每个节点 (node) 有零或者多个输入和零个或者多个输出, 表示一种操作 (operation) 的实例化
  - 流过图中正常的边 (从输出到输入) 的值叫做张量 (tensor)
  - 特殊的边, 叫做控制依赖 (control dependencies)



# 编程模型



# 编程模型

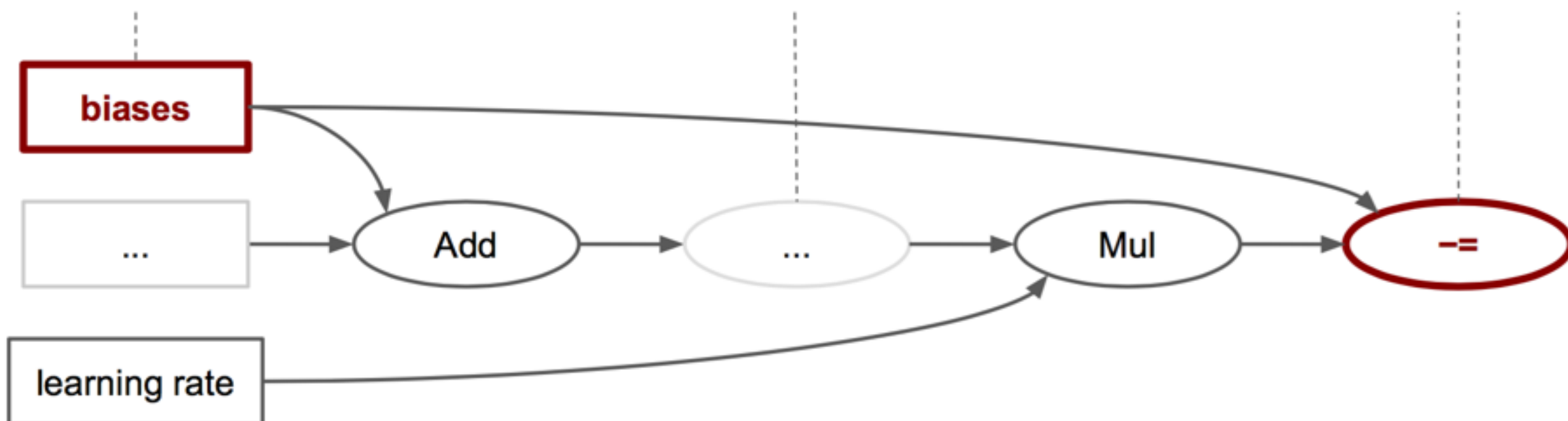


# 编程模型

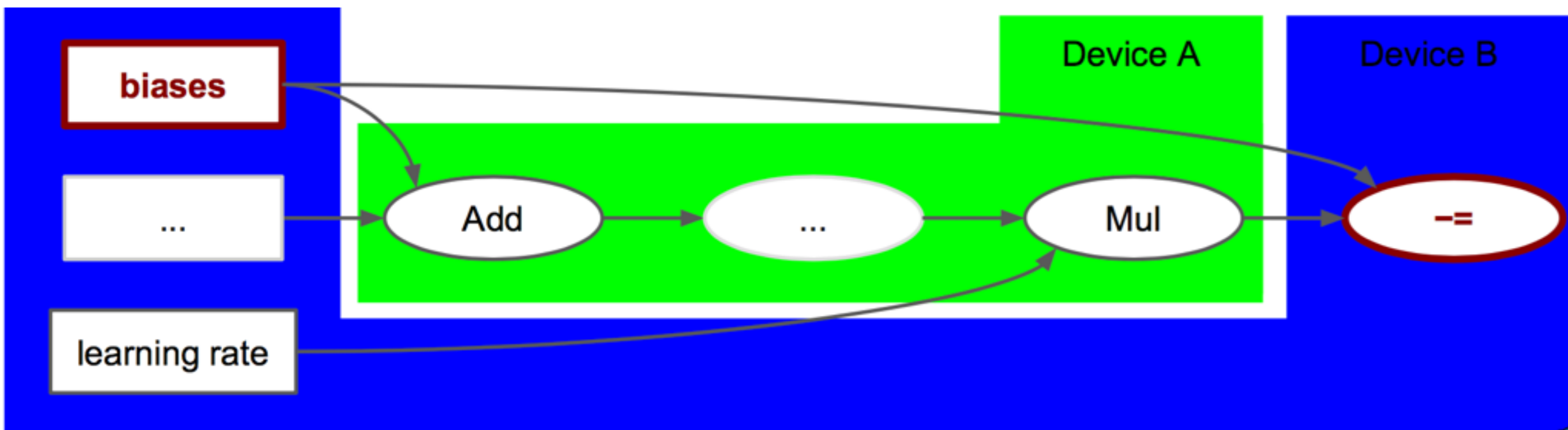
'Biases' is a variable

Some ops compute gradients

`--=` updates biases



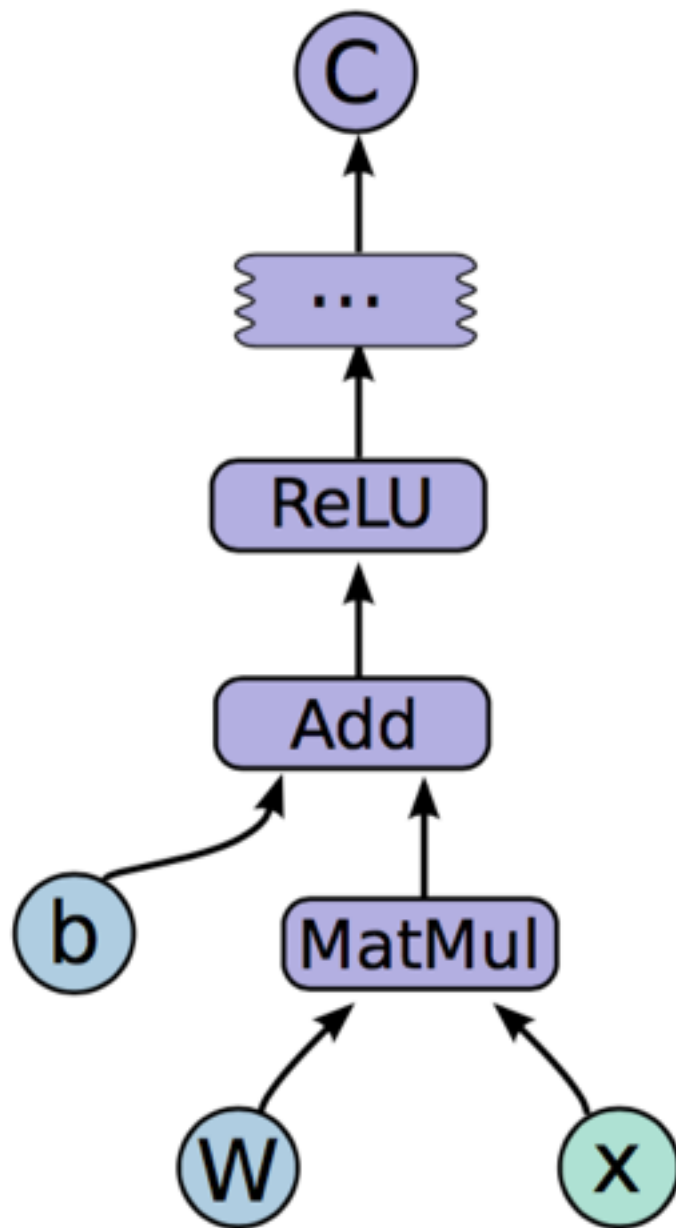
# 编程模型



# 基本用法

- 使用图 (graph) 来表示计算任务.
- 在被称之为 会话 (Session) 的上下文 (context) 中执行图.
- 使用 tensor 表示数据.
- 通过 变量 (Variable) 维护状态.
- 使用 feed 和 fetch 为任意操作输入和输出数据.

# 例子



```
import tensorflow as tf
```

```
b = tf.Variable(tf.zeros([100]))
```

```
W = tf.Variable(tf.random_uniform([784,100],-1,1))
```

```
x = tf.placeholder(name="x")
```

```
relu = tf.nn.relu(tf.matmul(W, x) + b)
```

```
C = [...]
```

```
# 100-d vector, init to zeroes
```

```
# 784x100 matrix w/rnd vals
```

```
# Placeholder for input
```

```
# Relu(Wx+b)
```

```
# Cost computed as a function
```

```
# of Relu
```

```
s = tf.Session()
```

```
for step in xrange(0, 10):
```

```
    input = ...construct 100-D input array ...
```

```
    result = s.run(C, feed_dict={x: input})
```

```
    print step, result
```

```
# Create 100-d vector for input
```

```
# Fetch cost, feeding x=input
```

# 弹性

- 通用计算基础设置
- 支持深度学习的核心库
- 也可用在其他的机器学习算法
- high performance computing (HPC)
- 与底层设备和硬件无关的抽象

# 可扩展

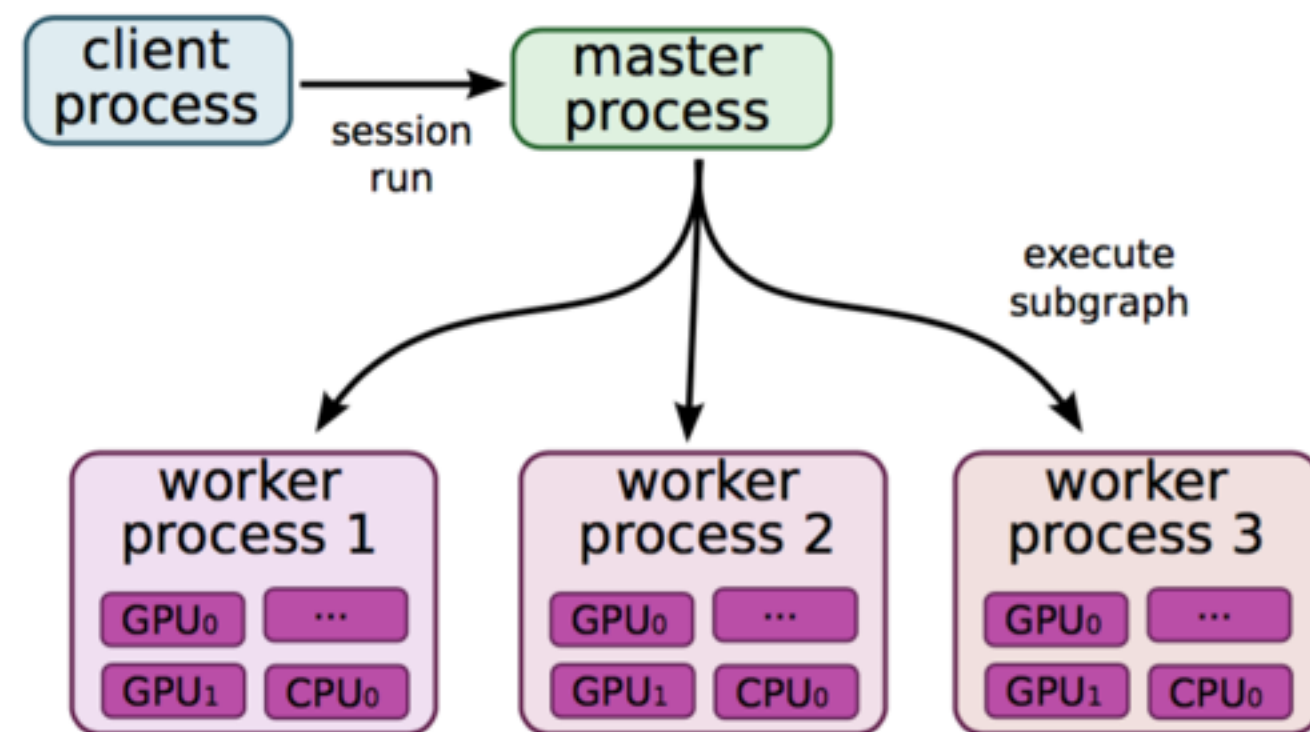
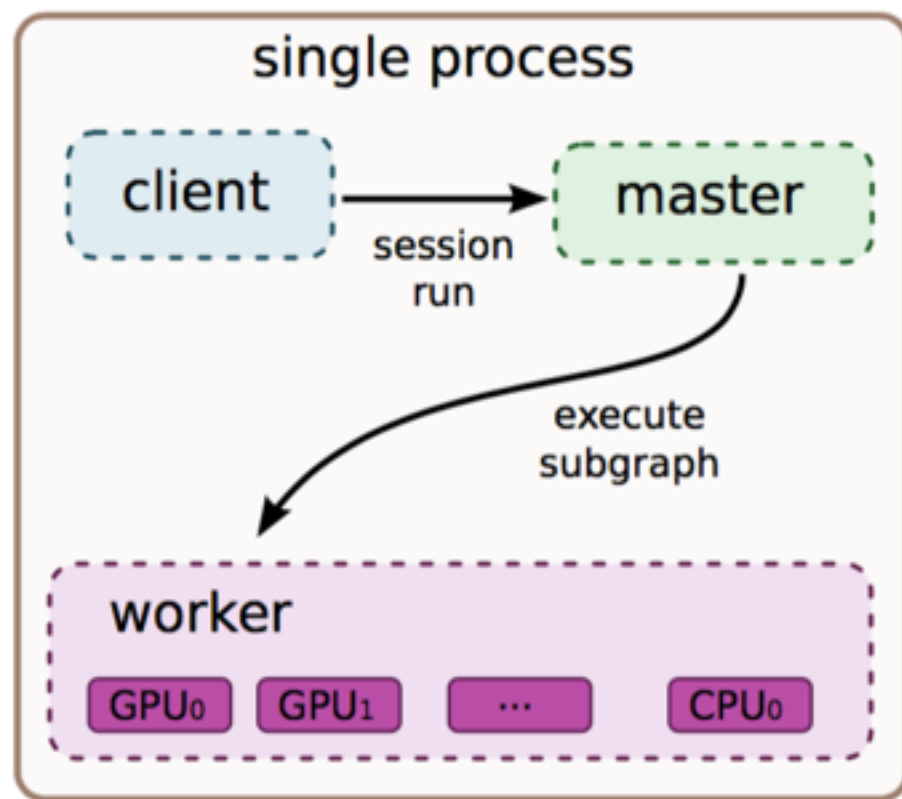
- 核心系统定义了一系列标准操作和kernel（操作在特定设备上的实现）
- 很容易定义一个新的operator和kernel



# 核心库支持操作

| 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                |

# 两种实现

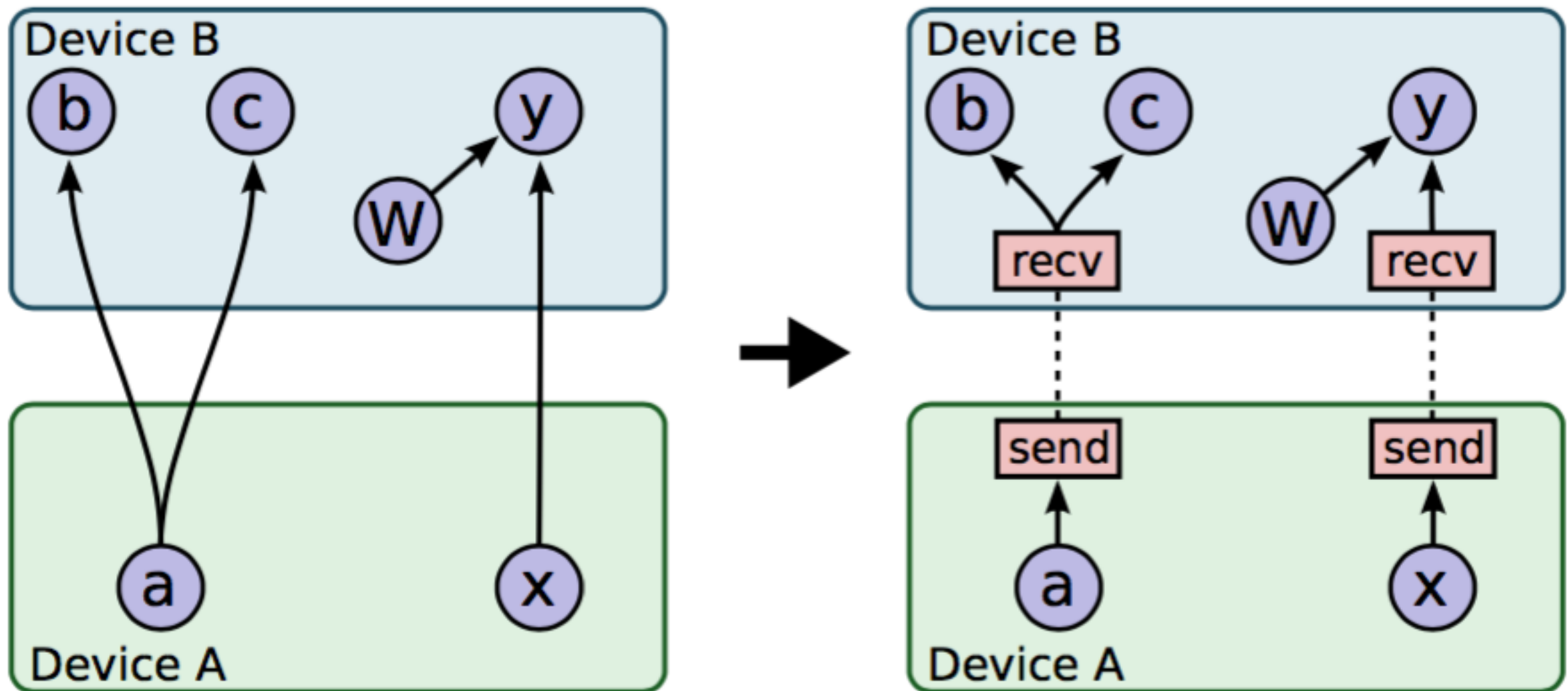


单机和分布式系统架构

# 设备分配

- 设备 (Devices)
  - TensorFlow 的计算核心
  - 每个 worker 负责一个或者多个设备
  - 每个设备有一个设备类型和一个名字
    - 由识别设备类型、设备索引, job和任务等这些标志组成
    - 例如: “/job:localhost/device:cpu:0 ”或者 “/job:worker/task:17/device:gpu:3”
  - 当前已实现了 CPU 和 GPU 的设备接口, 其他的设备类型也可以通过注册机制来实现
  - 每个设备对象负责管理分配和释放设备内存
- 分配算法
  - 代价模型
  - 贪心策略
  - 模拟选择

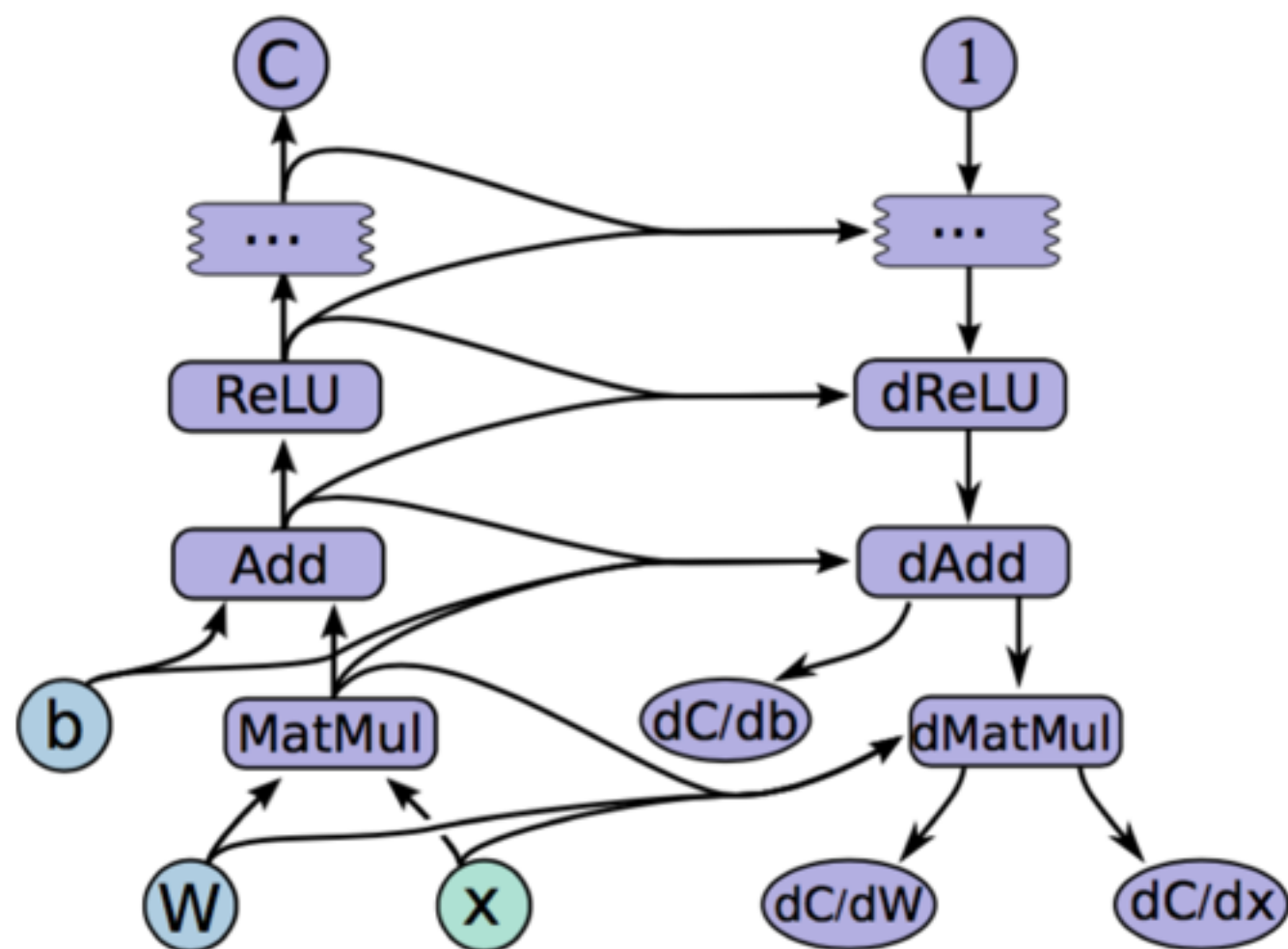
# 设备间通信



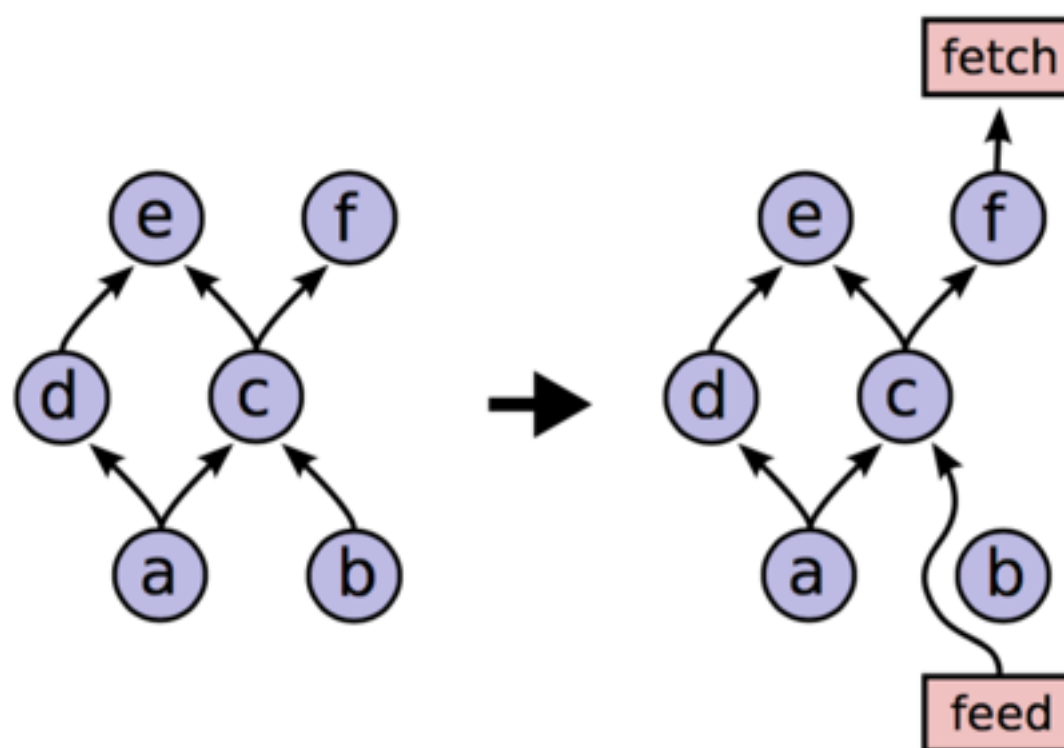
# 容错机制

- Send and Receive node间的通信错误
- master和worker进程之间的定期心跳检查。

# 梯度计算



# 部分执行



# 其他运行机制

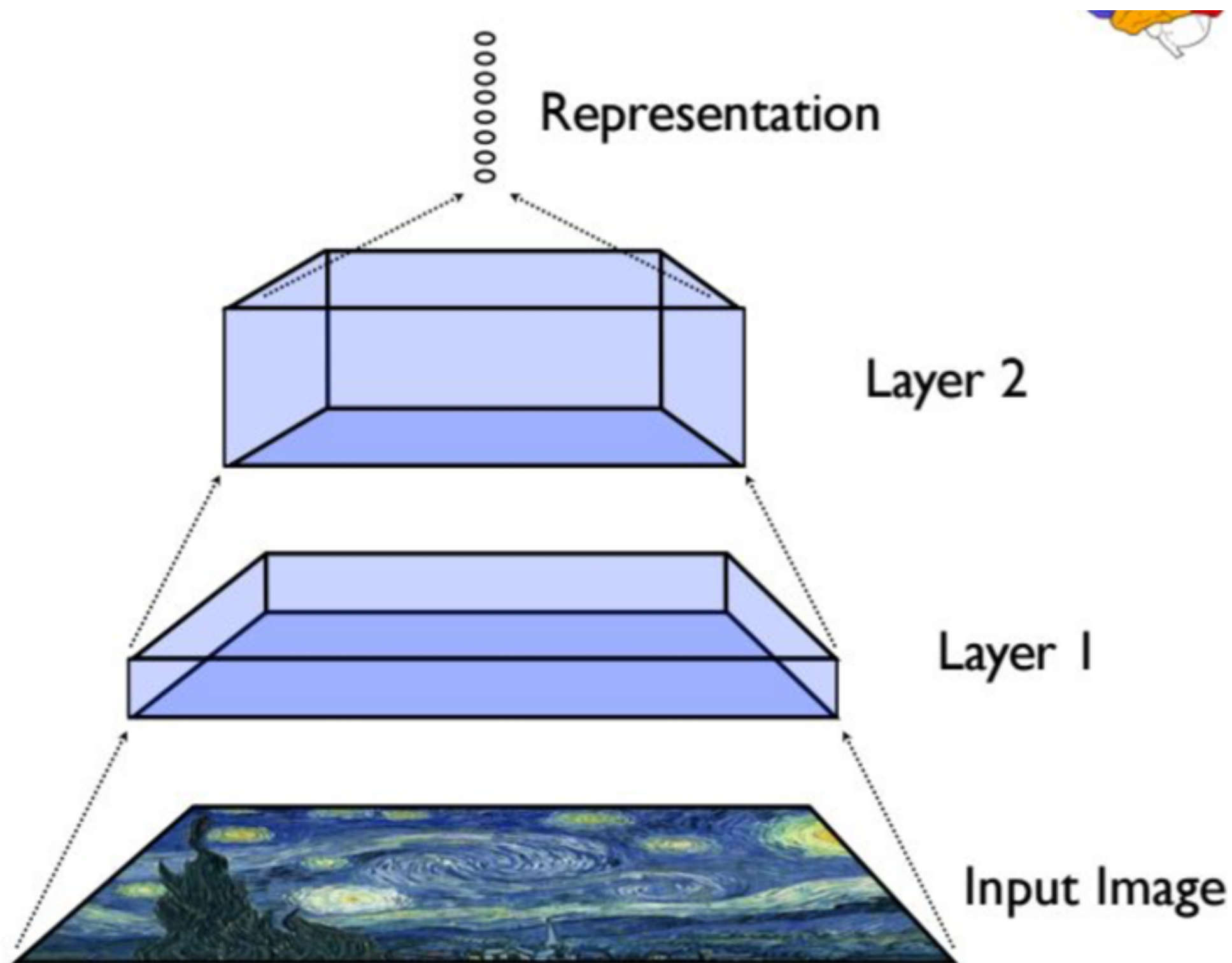
- 设备限制
- 控制流
- 输入操作
- 队列
- 容器



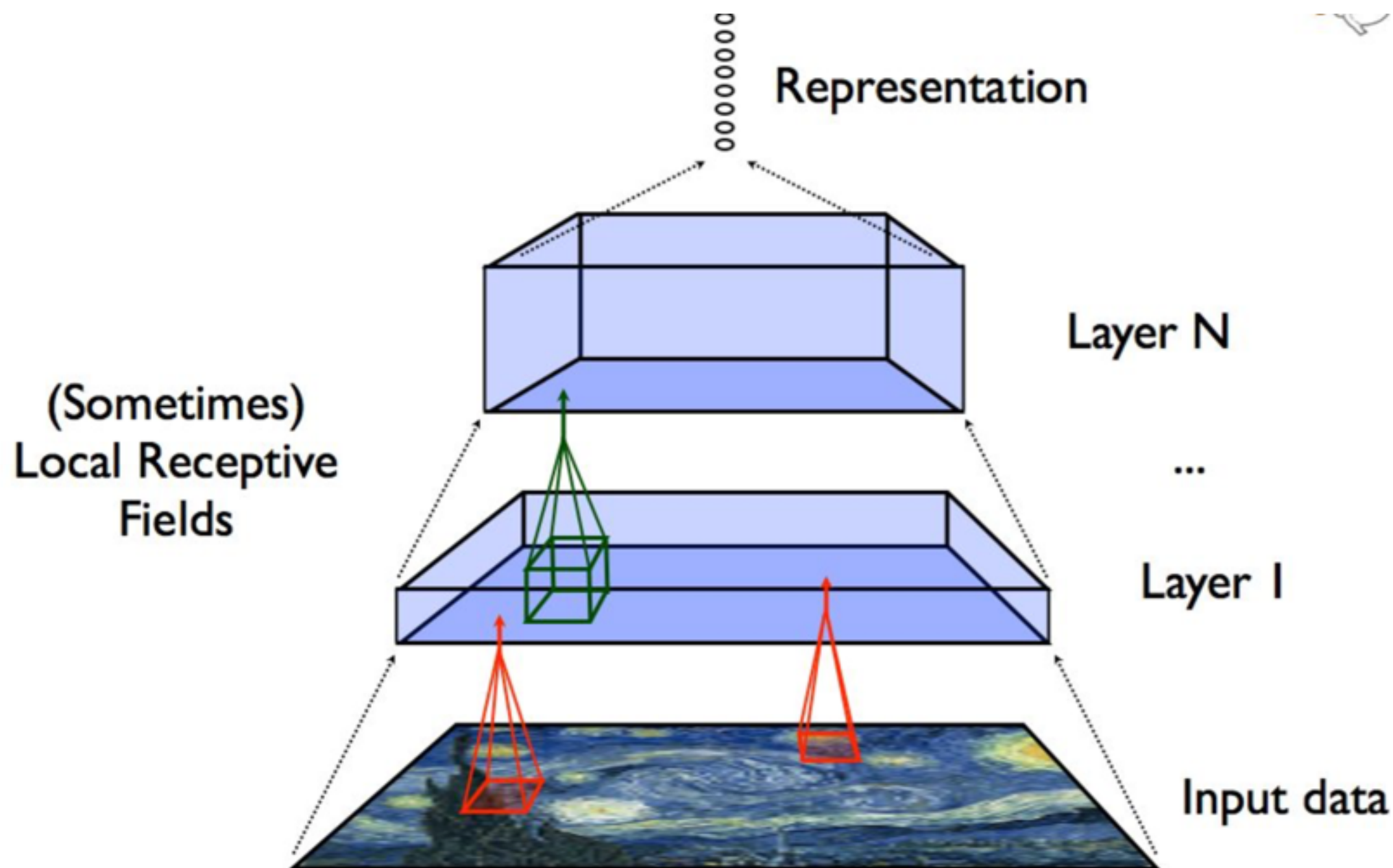
# 优化

- Common Subexpression Elimination
- Controlling Data Communication and Memory Usage
- Asynchronous Kernels
- Optimized Libraries for Kernel Implementations
- Lossy Compression

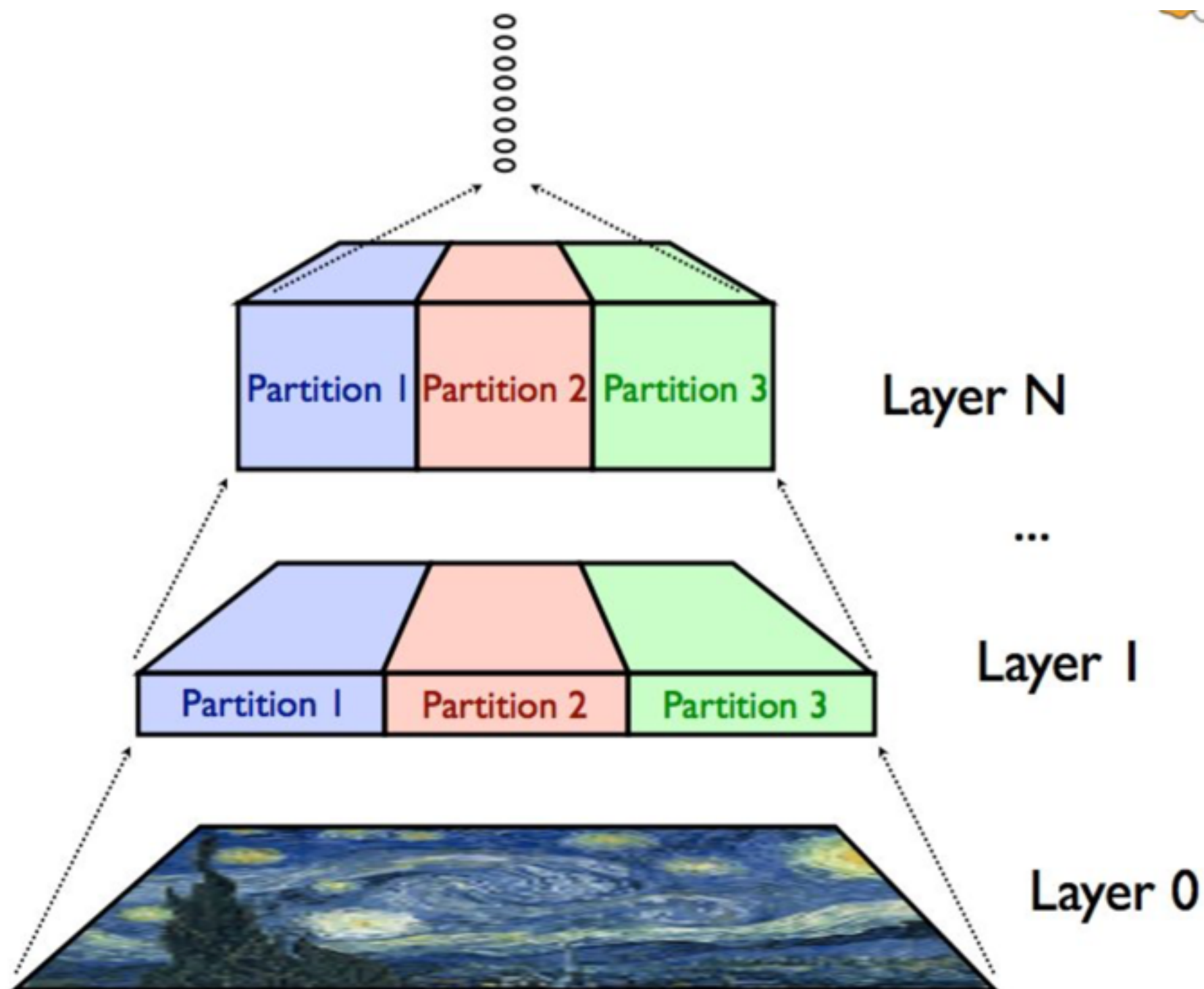
# 模型并行



# 模型并行



# 模型并行



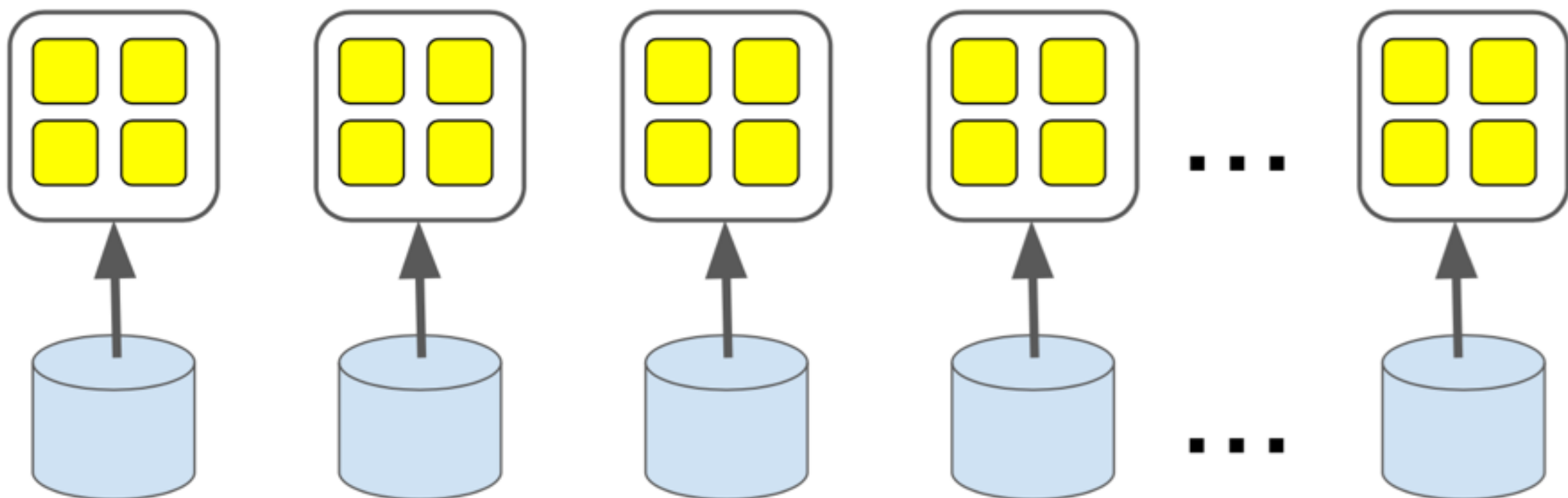
# 数据并行

Parameter Servers

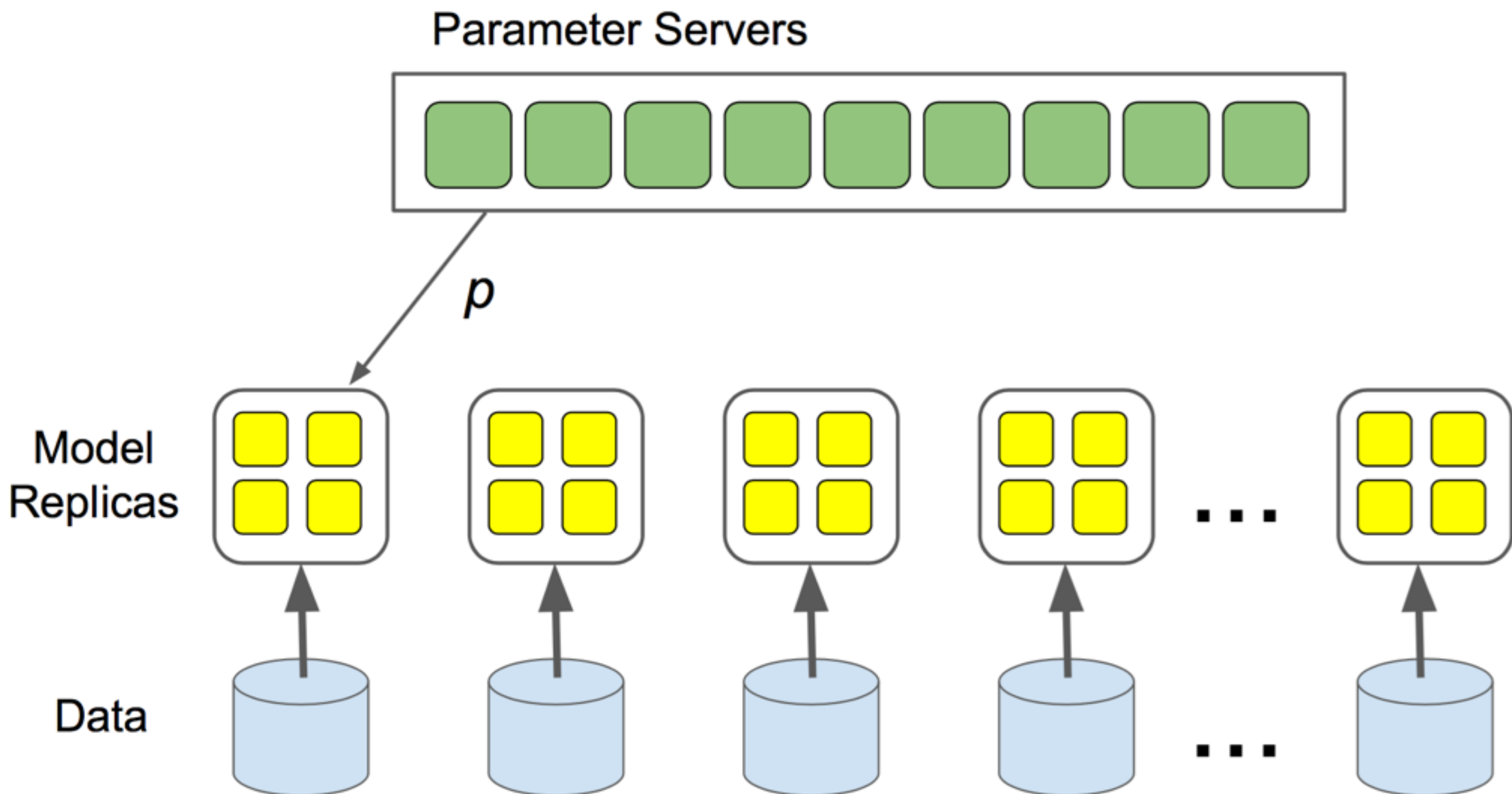


Model  
Replicas

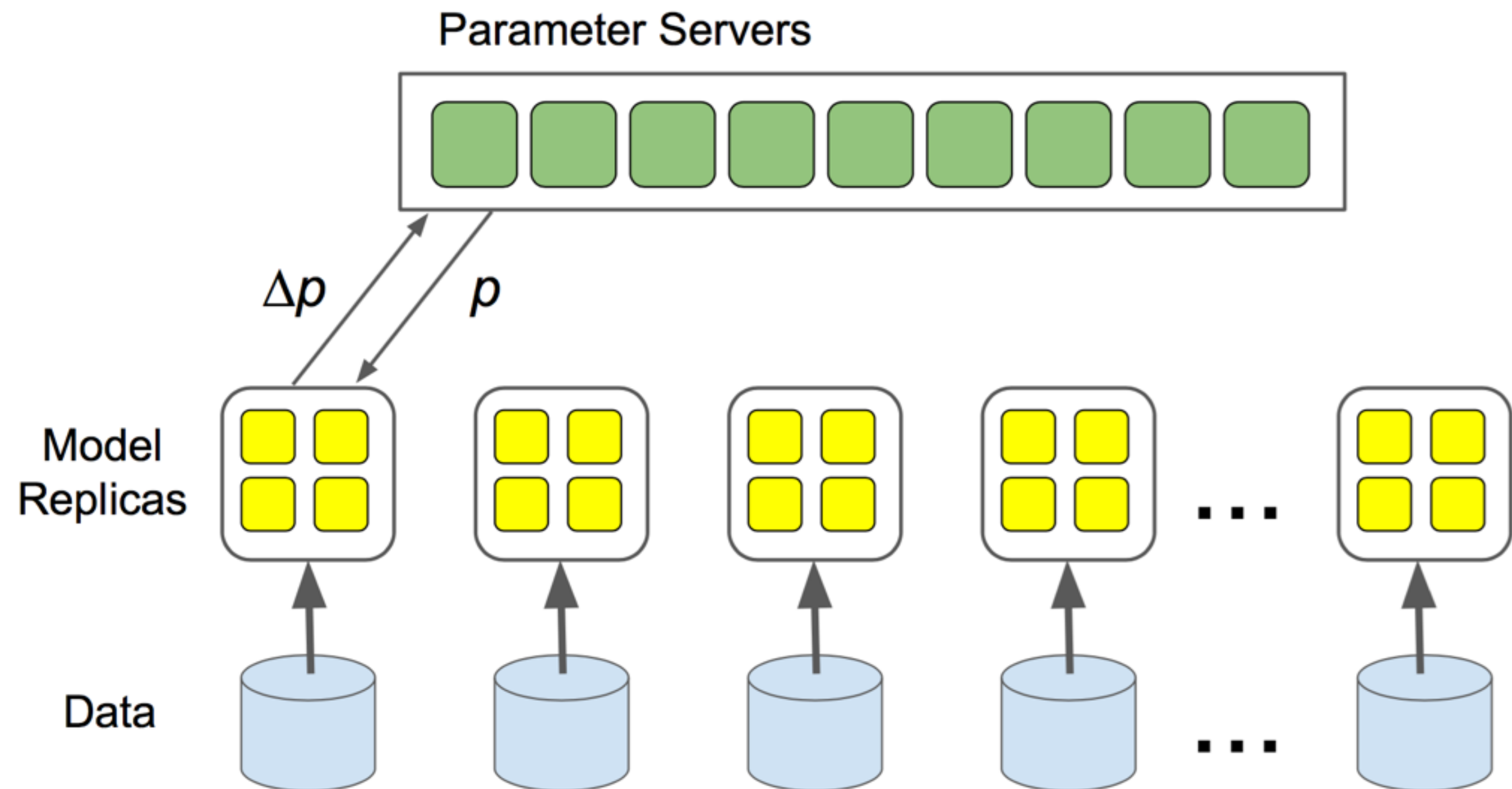
Data



# 数据并行

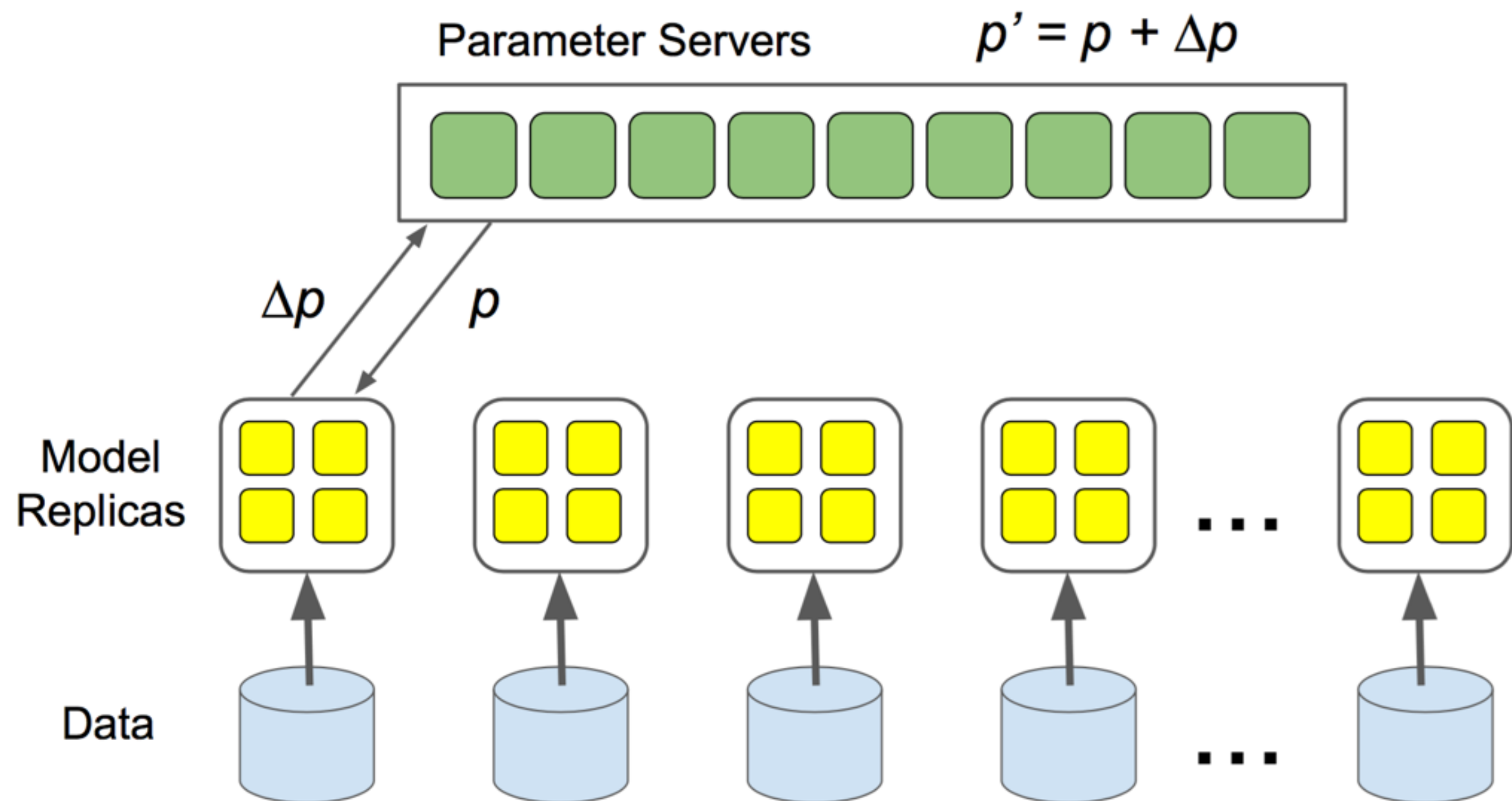


# 数据并行



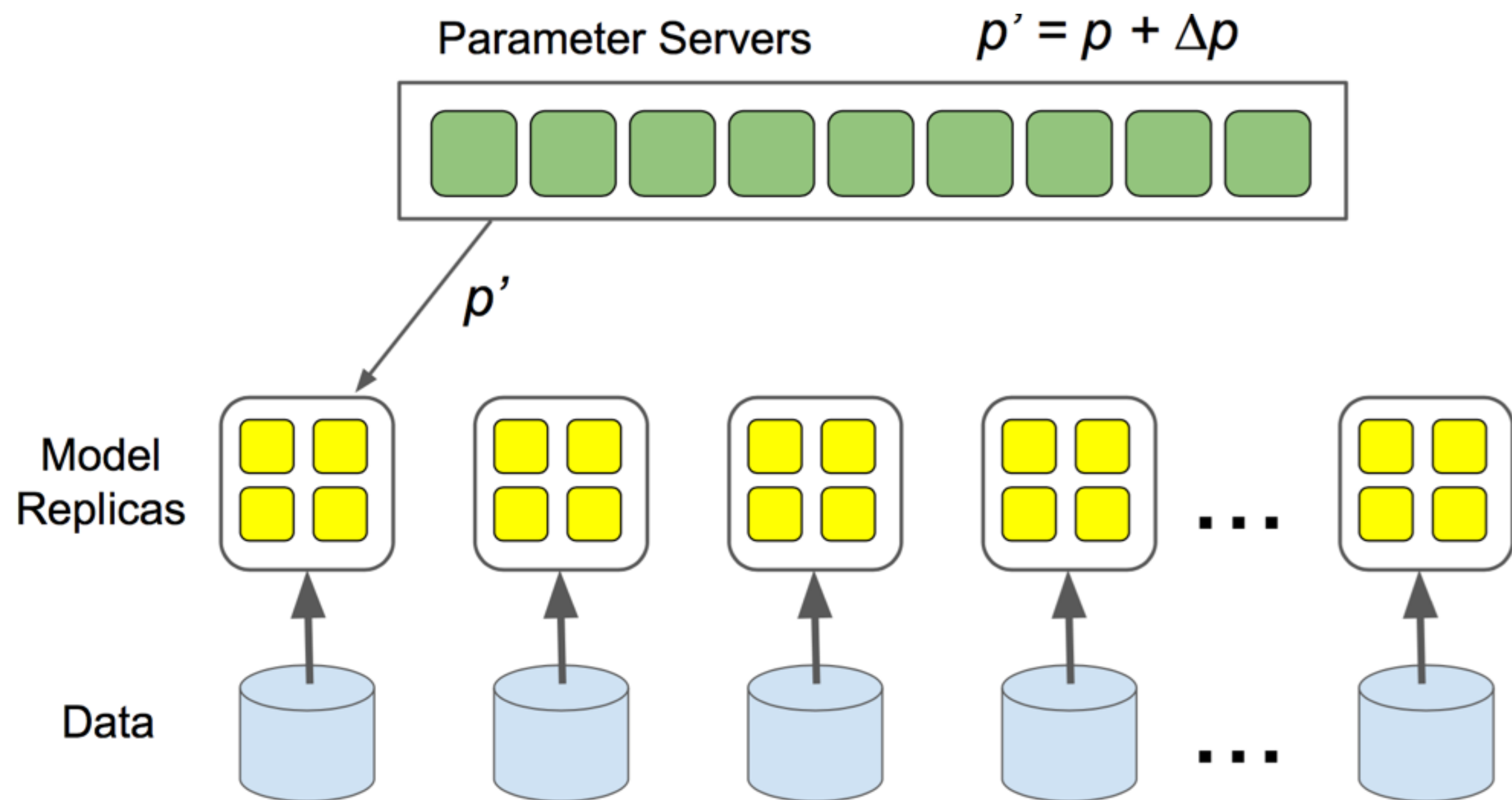


# 数据并行

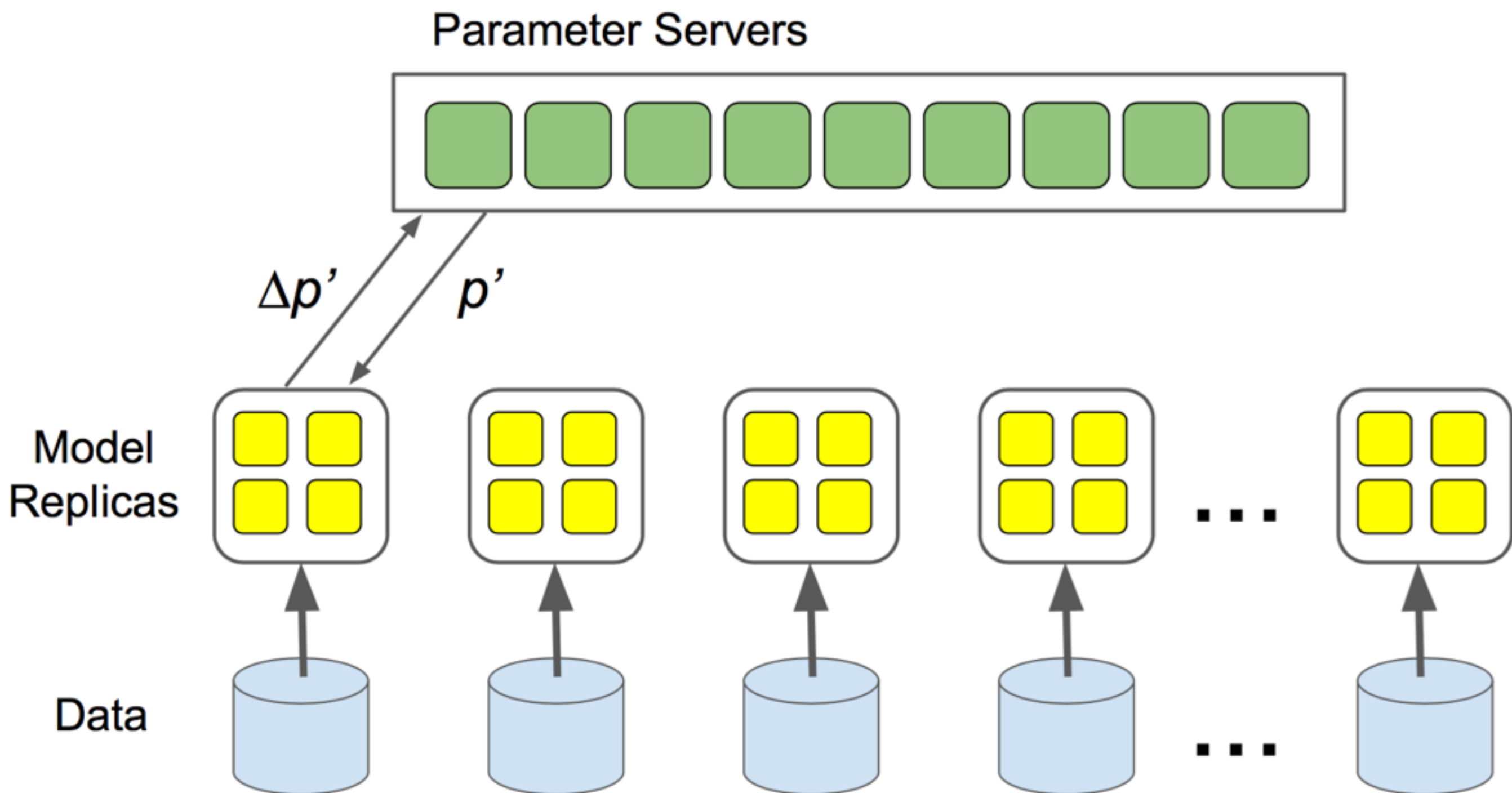




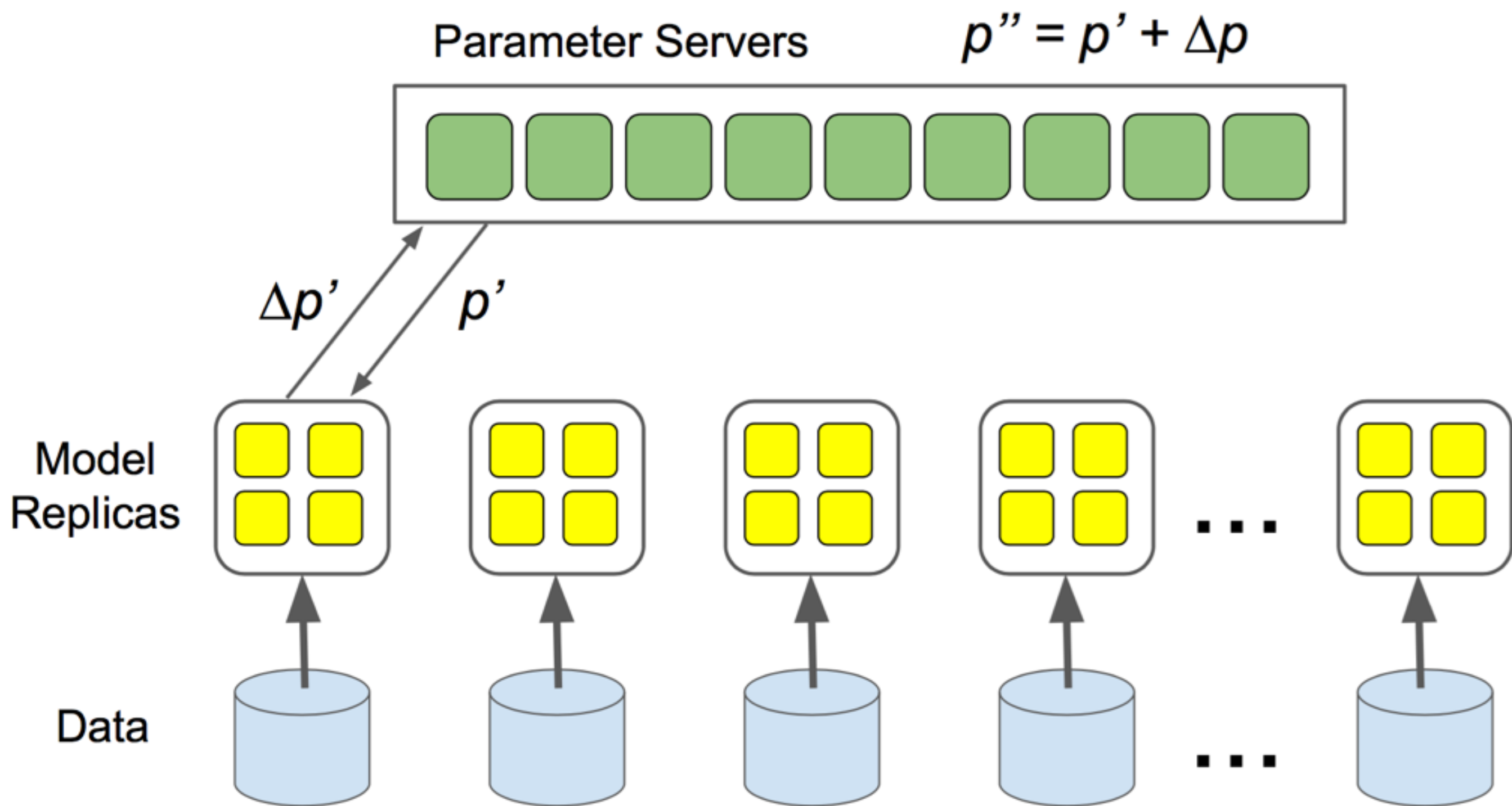
# 数据并行



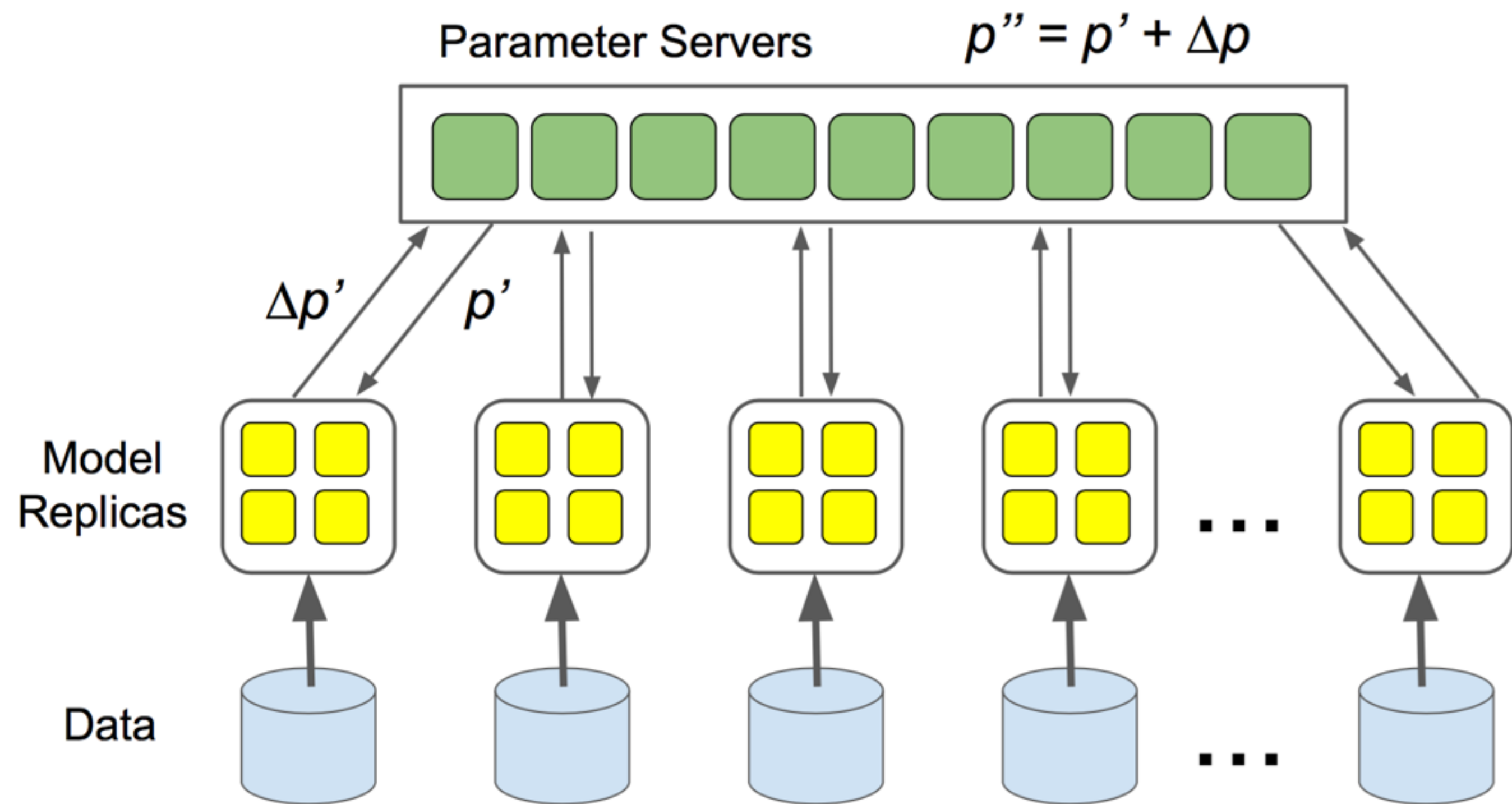
# 数据并行



# 数据并行



# 数据并行



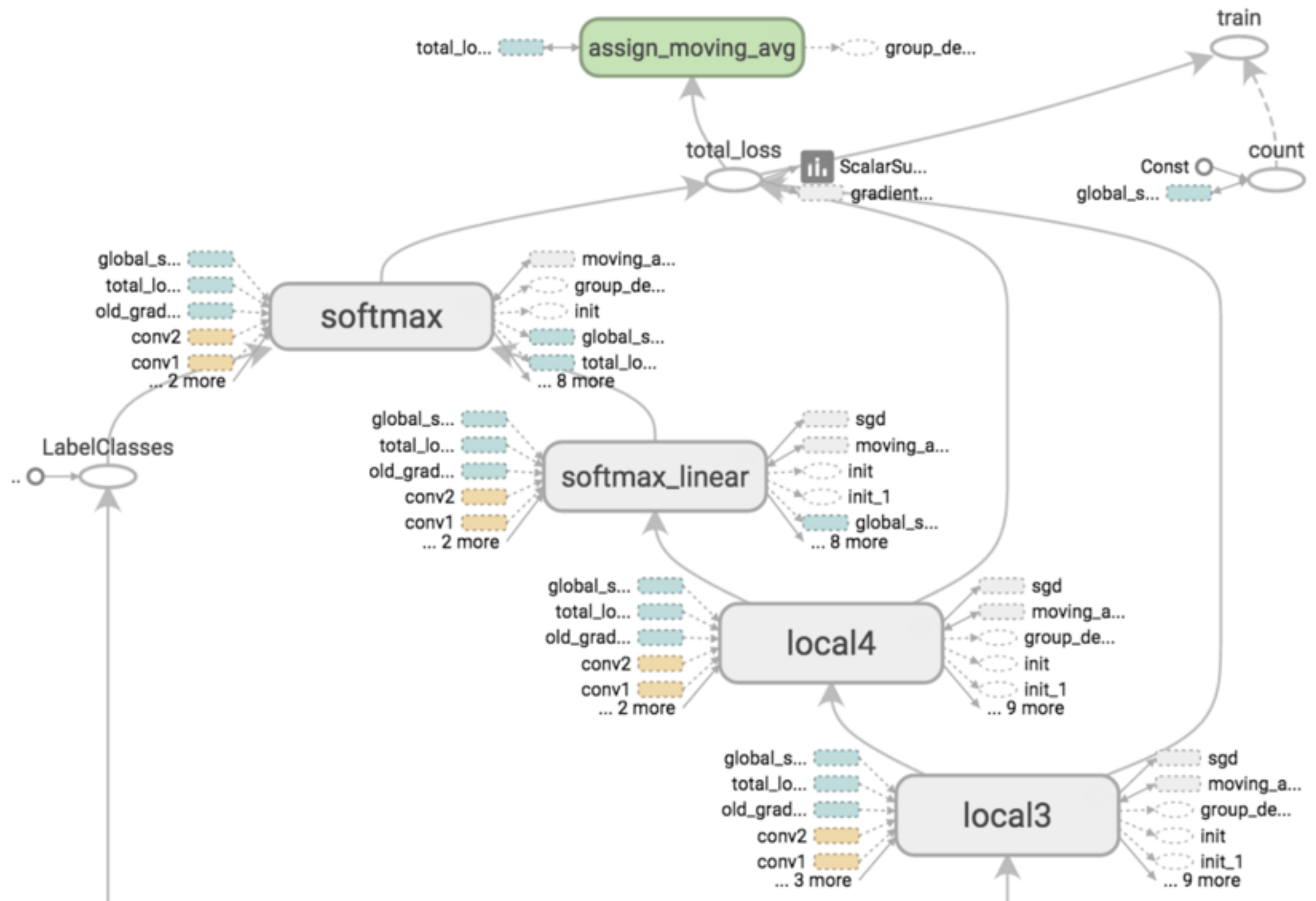
# 性能

- 官方当前未给出全面的性能评价
- 有人实测貌似比竞品慢

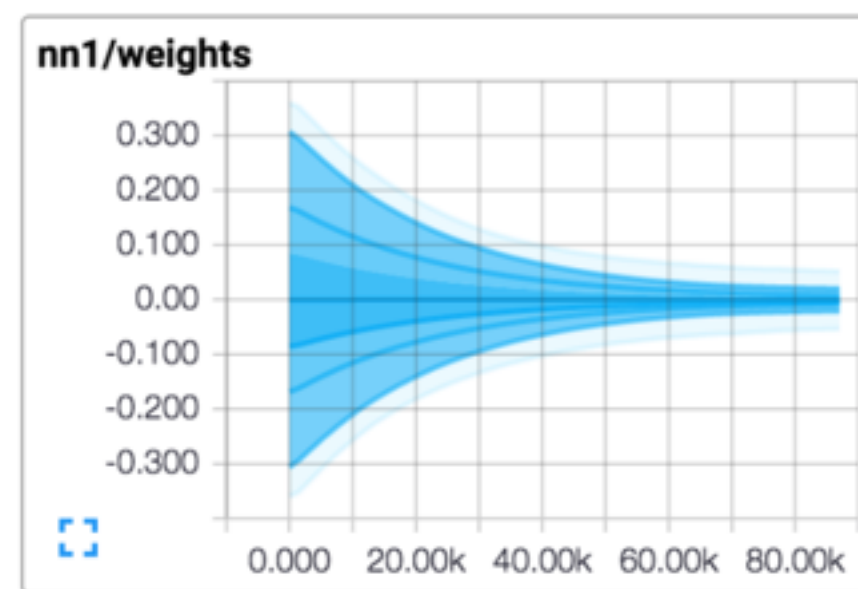
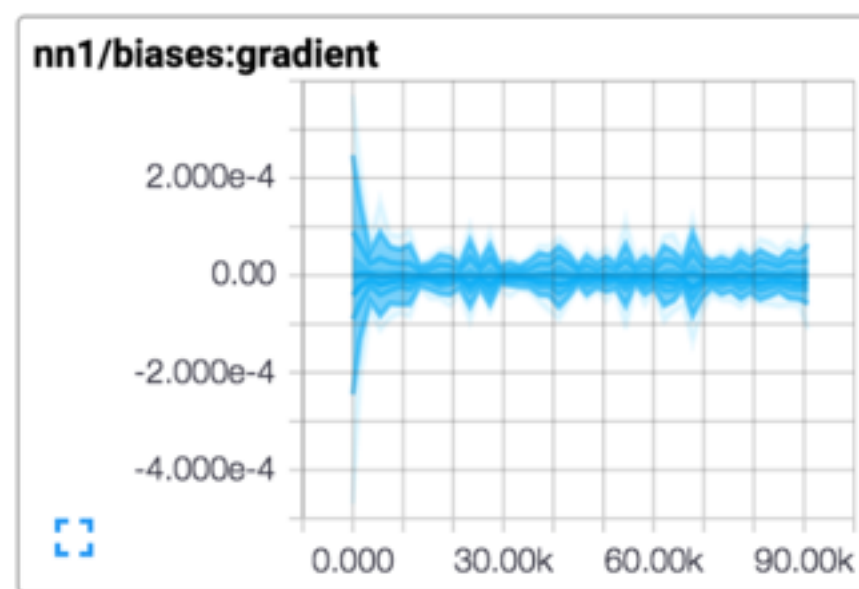
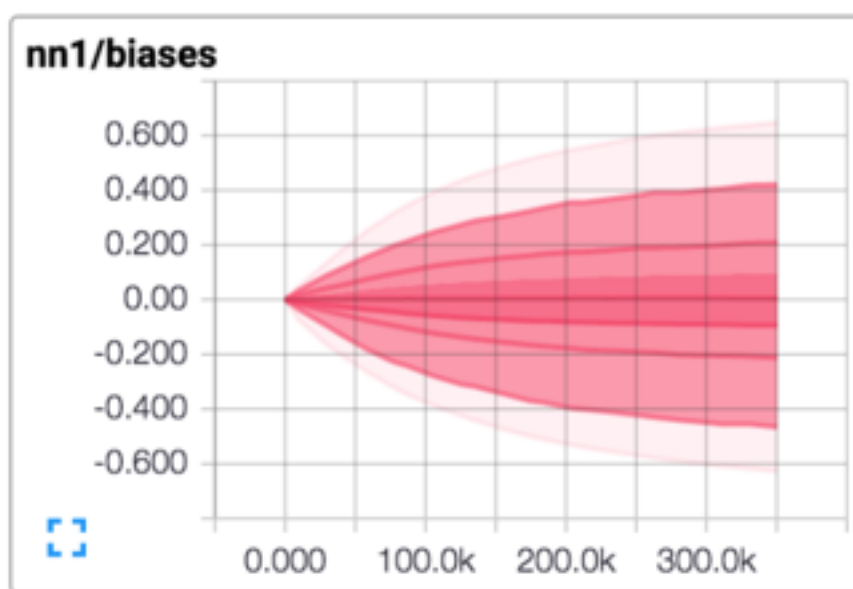
# 辅助工具

- TensorBoard
- Performance Tracing

# 计算可视化

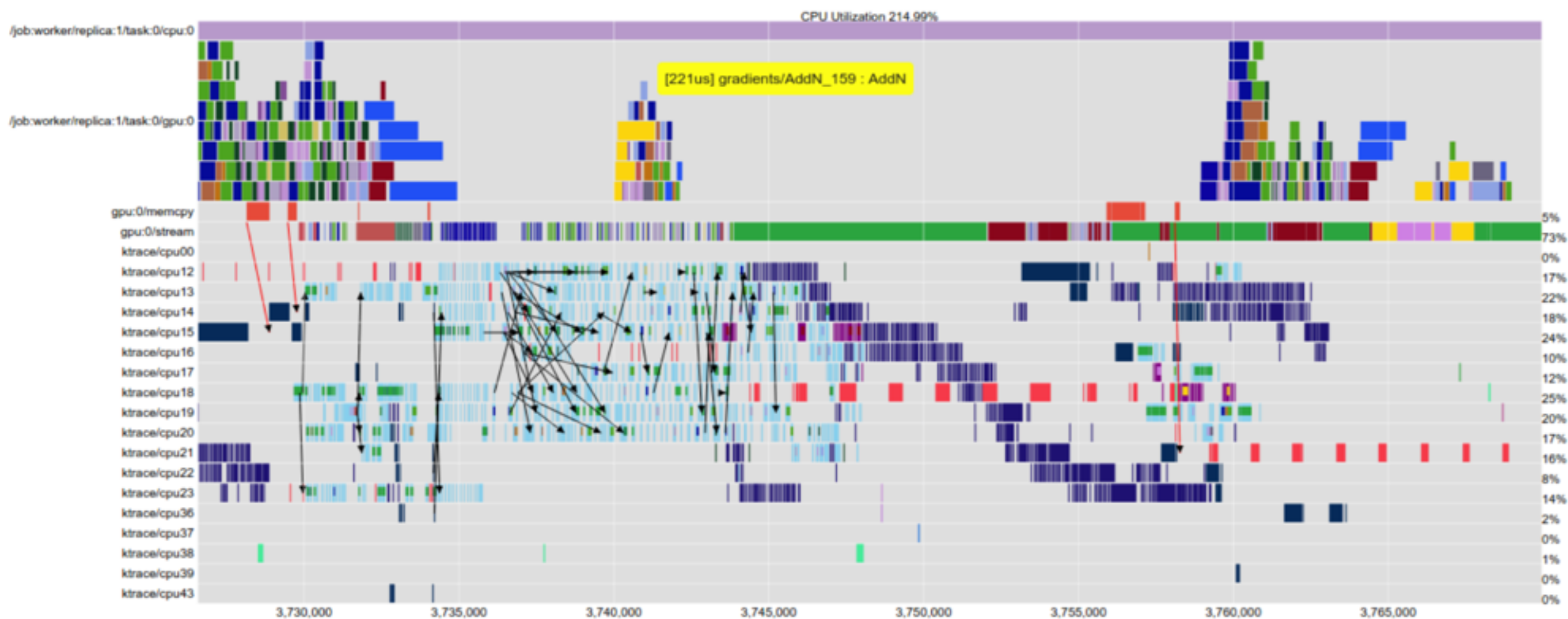


# 数据可视化





# 性能追踪



# 参考资料

- <http://www.tensorflow.org/>
- <https://github.com/tensorflow/tensorflow>
- <http://download.tensorflow.org/paper/whitepaper2015.pdf>
- <https://github.com/jikexueyuanwiki/tensorflow-zh>
- [白皮书翻译](#)

Q&A