

AiCon

全球人工智能与机器学习技术大会

深度学习框架技术剖析

袁进辉



北京一流科技有限公司
ONEFLOW TECHNOLOGY LIMITED

主办方 **Geekbang** 极客邦科技 **InfoQ** UQUE

TABLE OF CONTENTES

深度学习框架的定位

深度学习框架的最佳实践

深度学习框架当前技术焦点

主流深度学习框架点评

展望

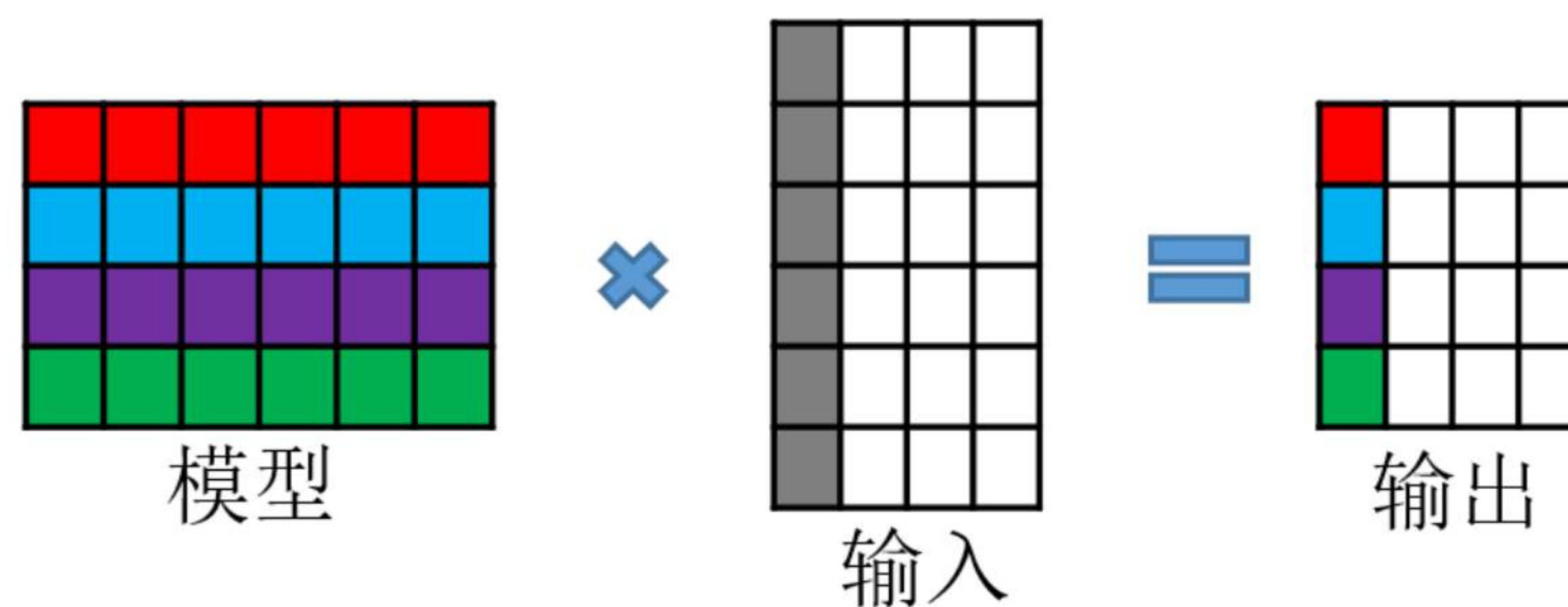
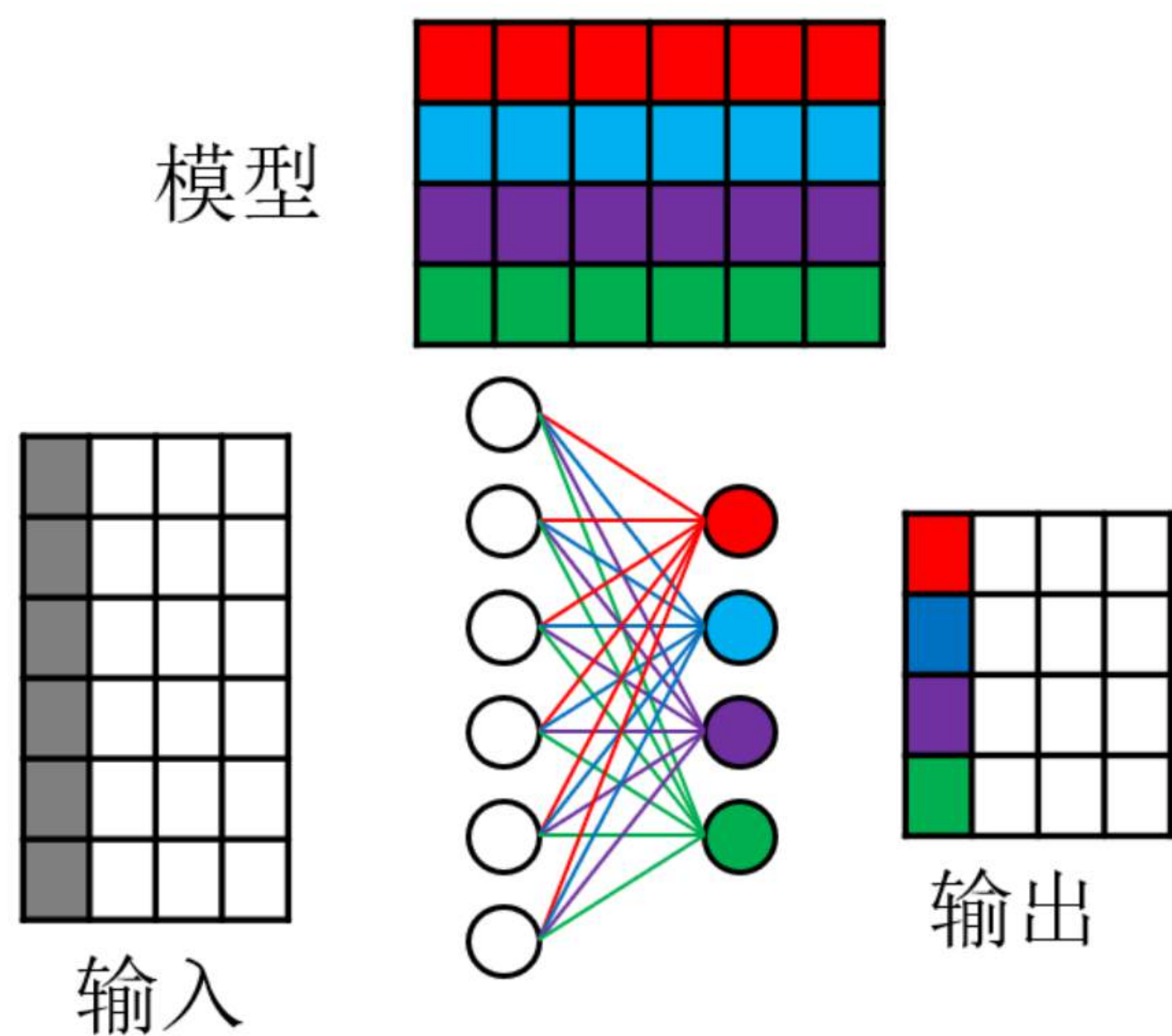
前提

- 深度学习当前已经取得巨大成功，未来还有旺盛生命力
- 算力在深度学习中扮演了重要角色（算力与算法）
- 深度学习软件和硬件至少一样重要（软件与硬件）
- 算力的瓶颈在分布式框架（宏观与微观）
- 深度学习软件解决编程不够快和程序运行不够快两个痛点

《深度学习平台技术演进》

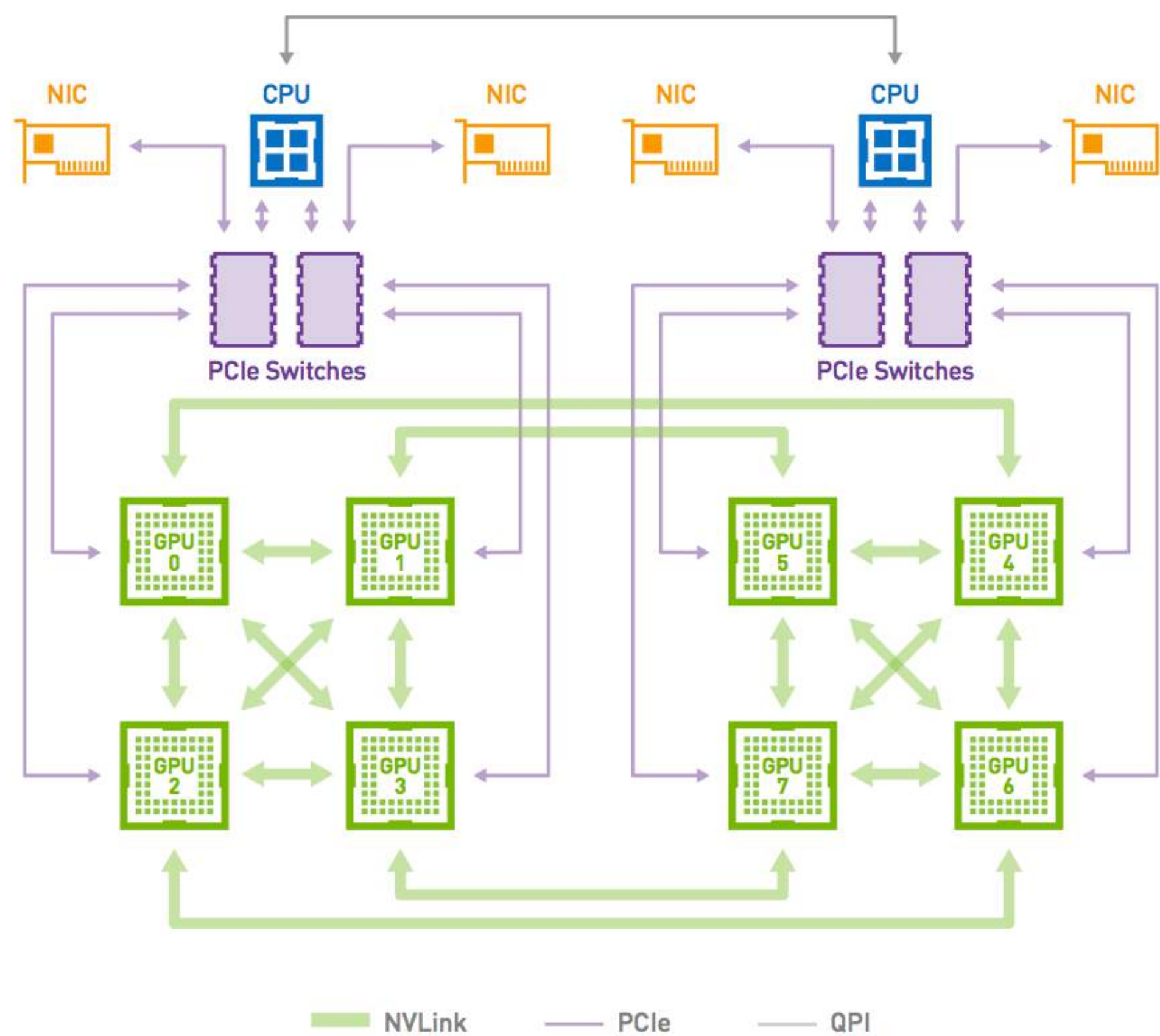


神经网络的矩阵表示



基于矩阵乘积表示的表示变换

高速互联实现硬件扩展性

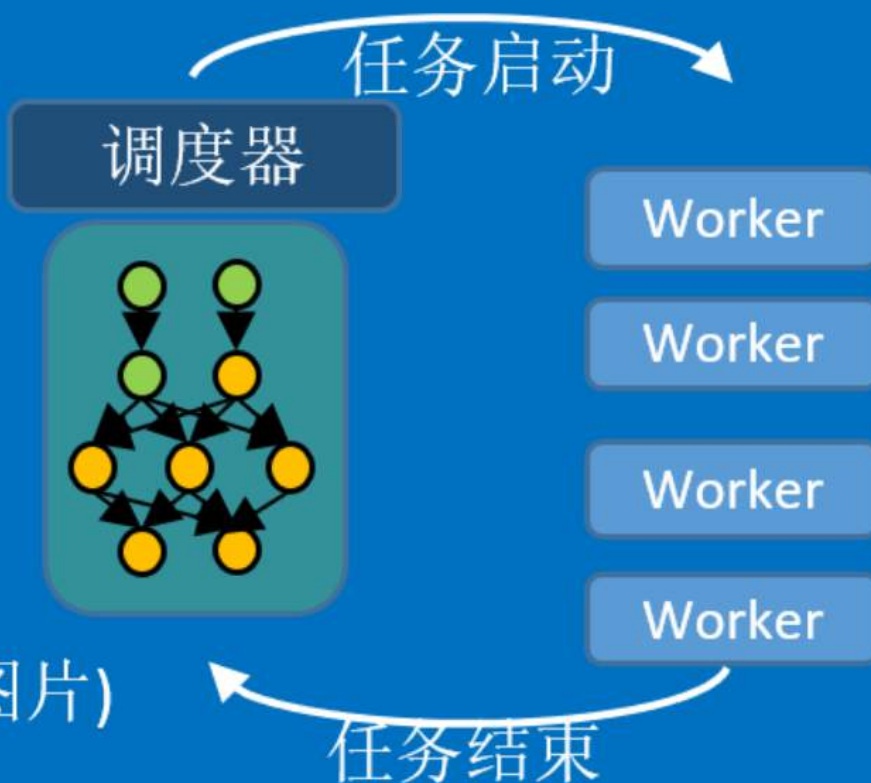


硬件越快，软件越难

算法: 从批处理到流式计算

AlexNet (One Weird Trick paper) - Input 128x3x224x224

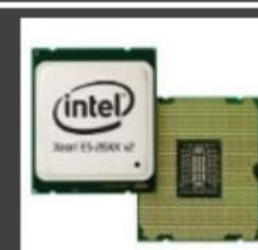
Library	Class	Time (ms)
CuDNN[R4]-fp16 (Torch)	cuda.SpatialConvolution	71
CuDNN[R4]-fp32 (Torch)	cuda.SpatialConvolution	81
Nervana-fp16	ConvLayer	92



单个GPU上小批次数据处理 (e.g. 128张图片)

新型硬件

CPU集群



6GB/s (InfiniBand)

0.7TFLOPS



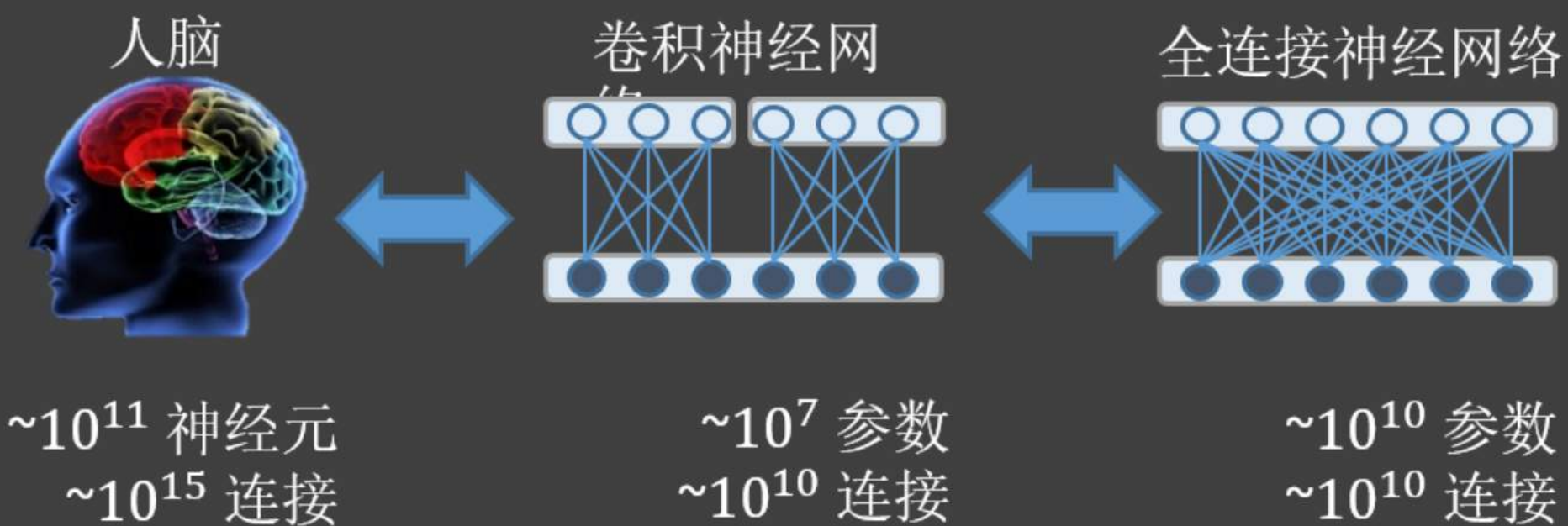
6GB/s with PCI-E

6GB/s with InfiniBand

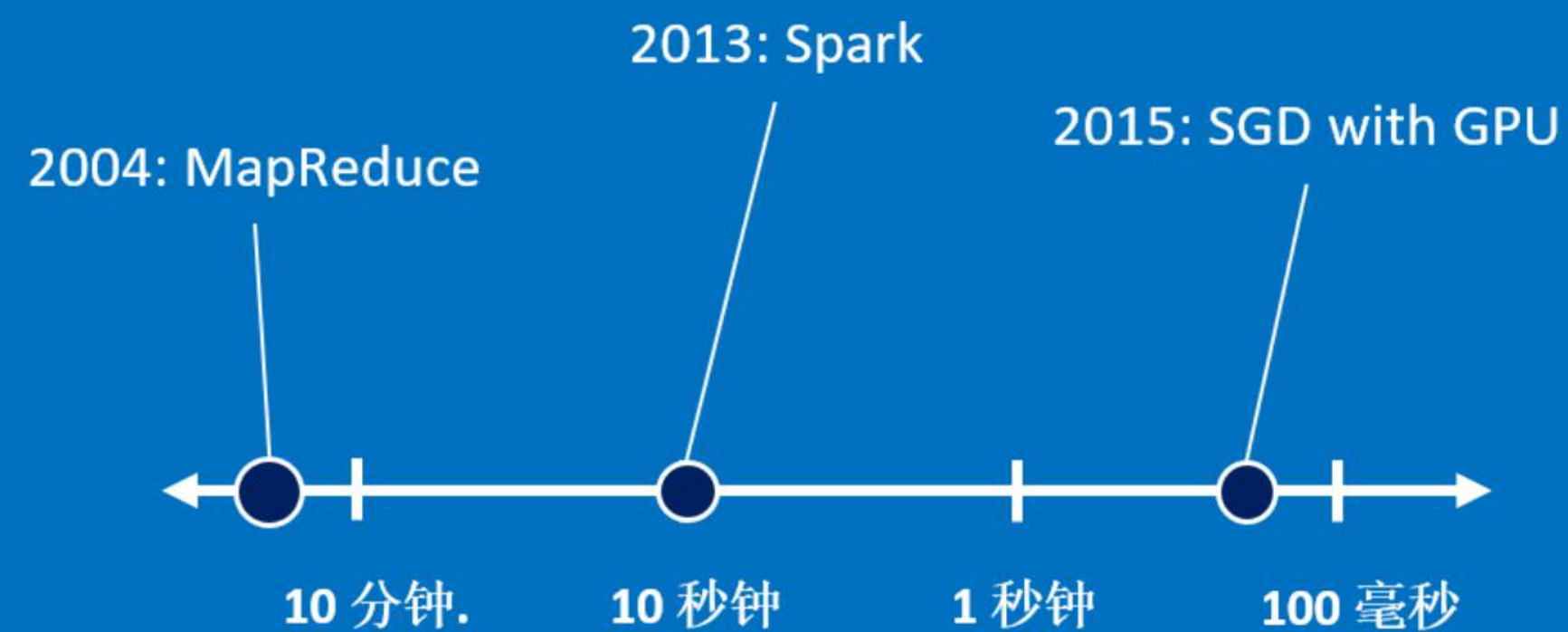
GPU 集群

10TFLOPS

大数据和大模型



任务粒度



深度学习软件平台的定位



TABLE OF CONTENTES

深度学习框架的定位

深度学习框架的最佳实践

深度学习框架当前技术焦点

主流深度学习框架点评

展望

控制流与数据流

$a := x + y$
 $b := a \times a$
 $c := 4 - a$

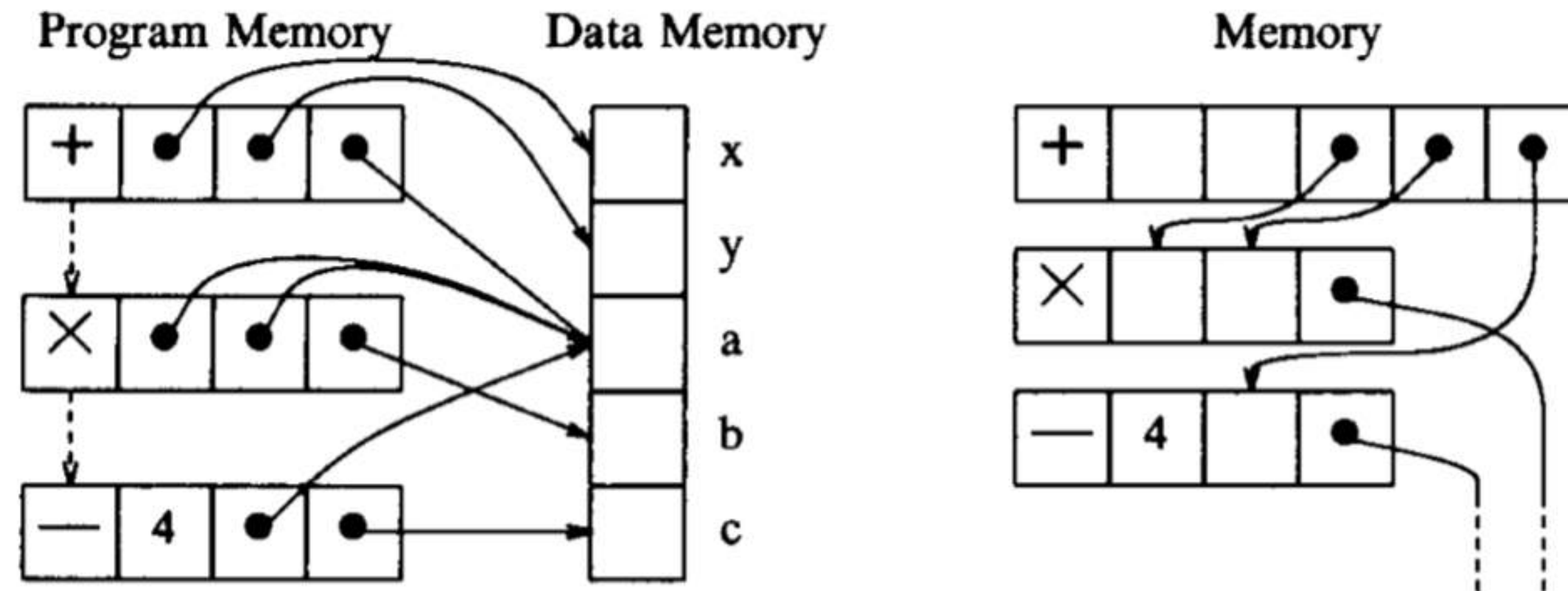
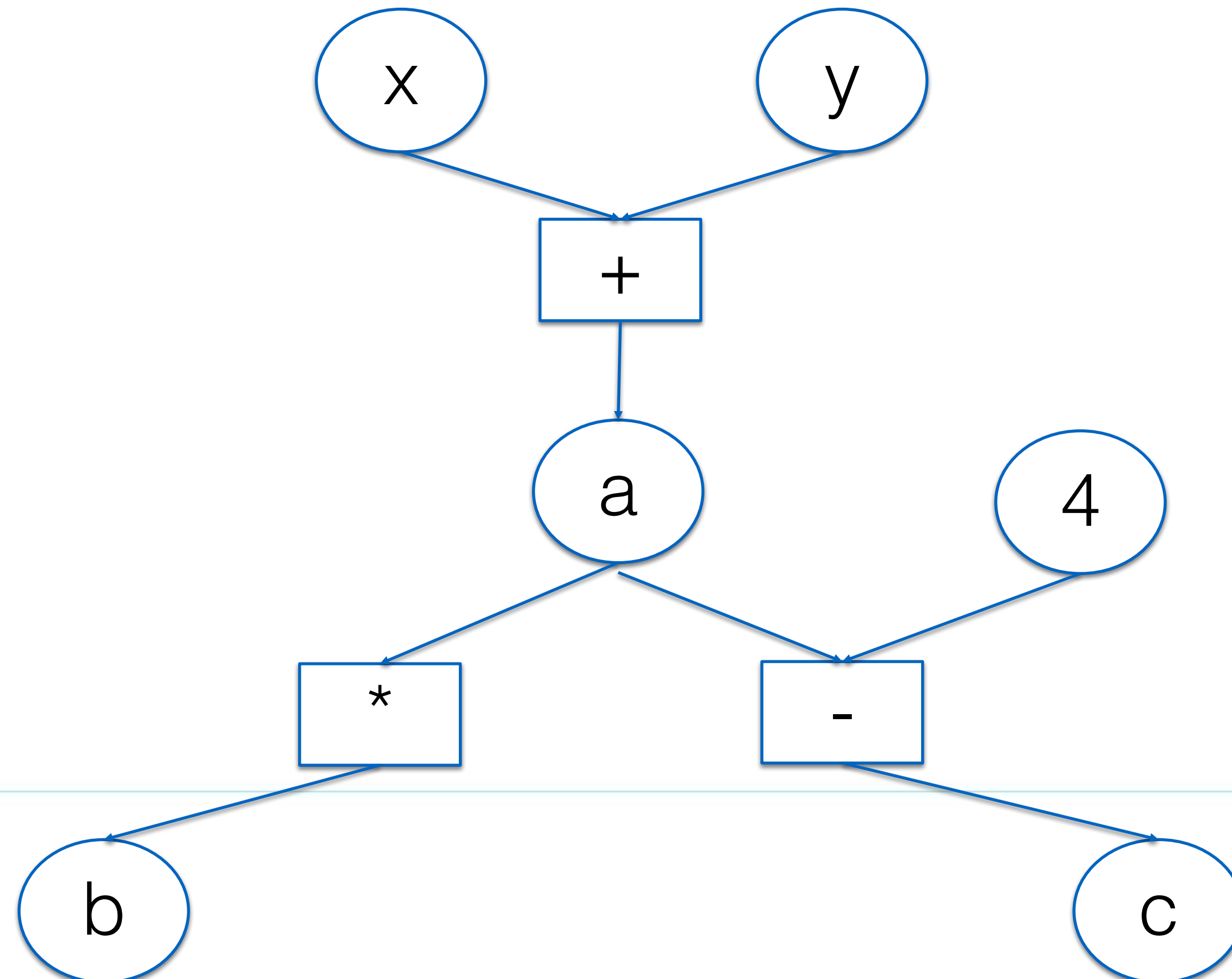


Figure 2. A comparison of control flow and dataflow programs. On the left a control flow program for a computer with memory-to-memory instructions. The arcs point to the locations of data that are to be used or created. Control flow arcs are indicated with dashed arrows; usually most of them are implicit. In the equivalent dataflow program on the right only one memory is involved. Each instruction contains pointers to all instructions that consume its results.

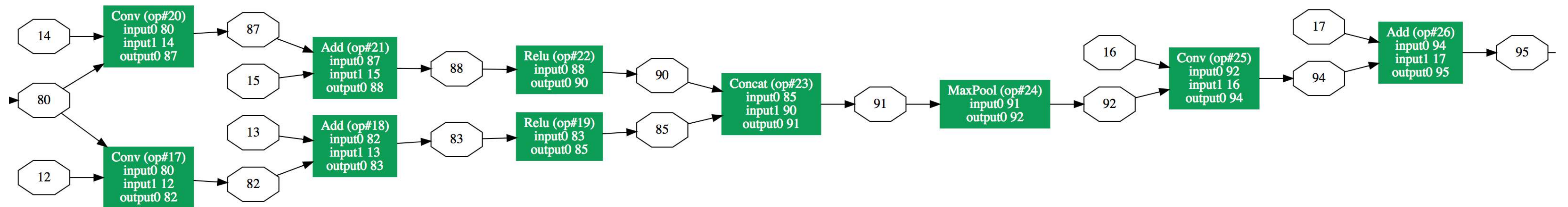
数据流：有向无环图

- 显式描述所有并行机会；简洁优雅的执行引擎



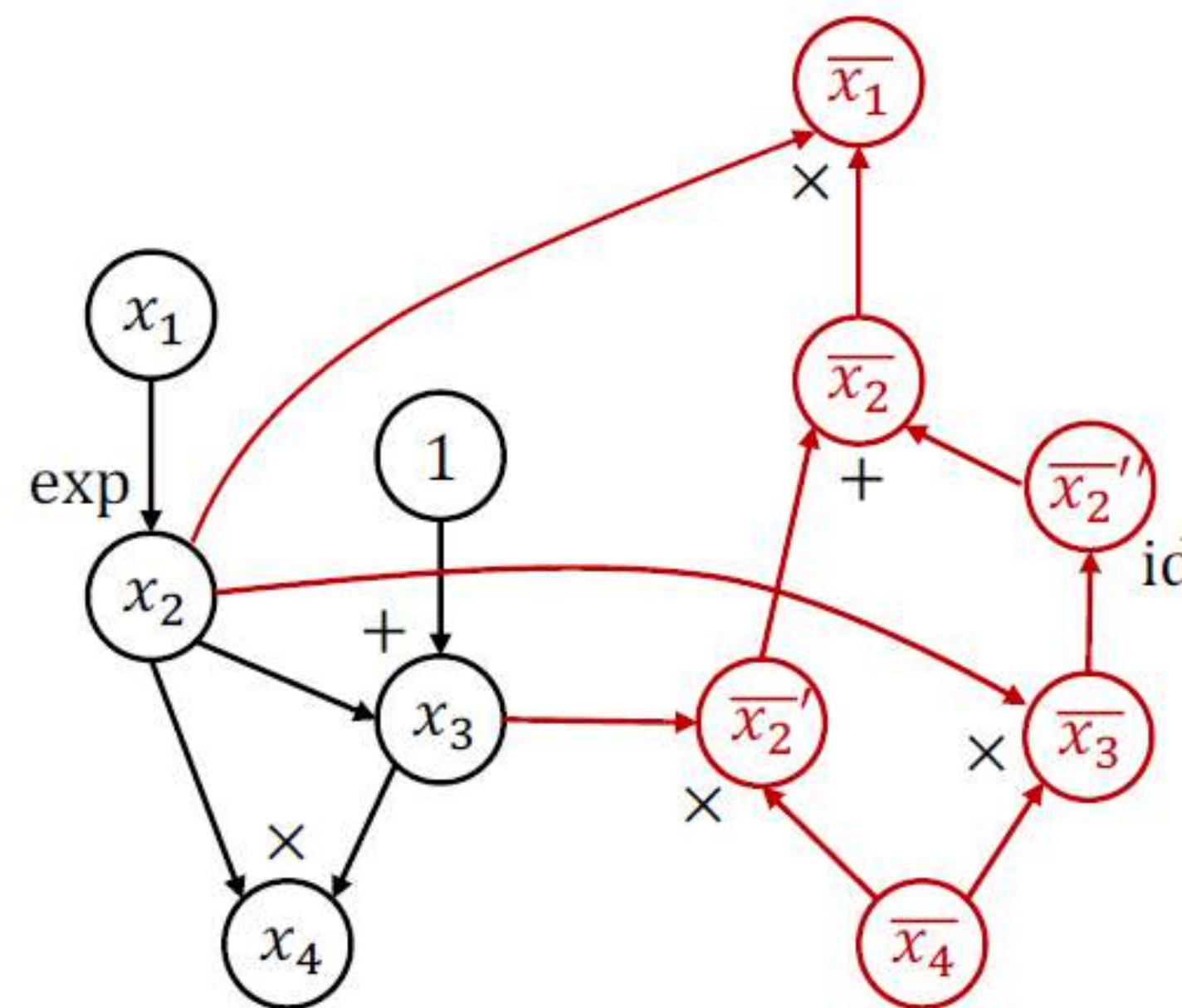
操作符与张量

- 操作符：常见神经网络基本运算（Op/Kernel, 粒度问题）
- 数据：张量（Eigen, Mshadow）



自动梯度计算 (autograd)

- 用户只须声明前向计算图，系统自动推导后向计算图
- 链式法则
- 依赖前向计算结果
- 多个消费者依赖同一数据时较tricky

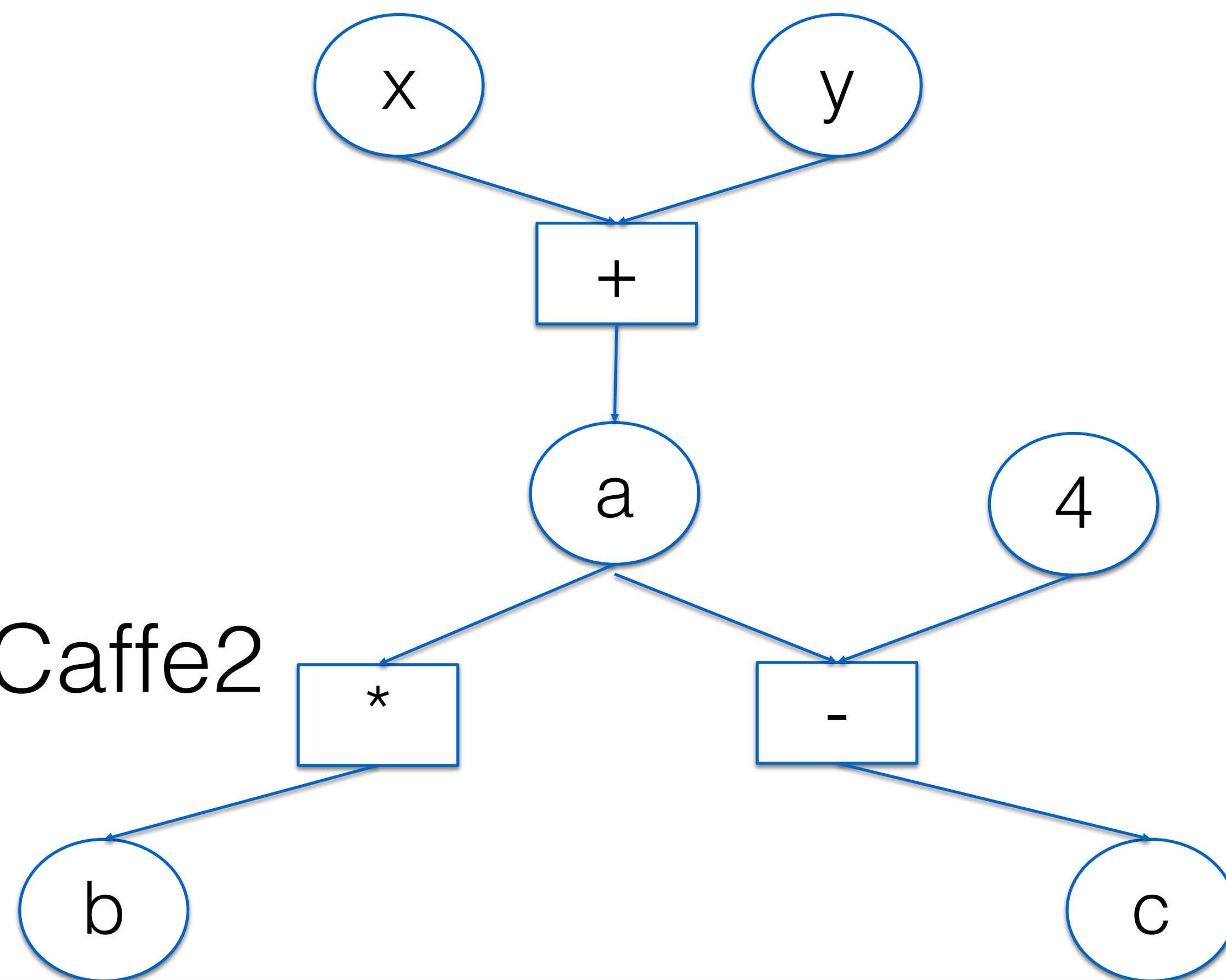


图重写（优化）

- 编译器技术：Pattern match 或 Pass
- 无用代码消除（Dead code elimination）
- 公共子表达式（Common sub expression）
- 操作符融合（Operator fusion）
- 类型/形状推导（Shape inference）
- 内存优化（Memory planning）

执行引擎 (Execution Engine)

- 拓扑序
- CPU调度, GPU执行
- 所有大数据引擎的内核
- 建议观摩: Tensorflow, MXNet, Caffe2



编程接口

Numpy

```
import numpy as np
np.random.seed(0)

N, D = 3, 4

x = np.random.randn(N, D)
y = np.random.randn(N, D)
z = np.random.randn(N, D)

a = x * y
b = a + z
c = np.sum(b)

grad_c = 1.0
grad_b = grad_c * np.ones((N, D))
grad_a = grad_b.copy()
grad_z = grad_b.copy()
grad_x = grad_a * y
grad_y = grad_a * x
```

TensorFlow

```
import numpy as np
np.random.seed(0)
import tensorflow as tf

N, D = 3, 4

with tf.device('/gpu:0'):
    x = tf.placeholder(tf.float32)
    y = tf.placeholder(tf.float32)
    z = tf.placeholder(tf.float32)

    a = x * y
    b = a + z
    c = tf.reduce_sum(b)

grad_x, grad_y, grad_z = tf.gradients(c, [x, y, z])

with tf.Session() as sess:
    values = {
        x: np.random.randn(N, D),
        y: np.random.randn(N, D),
        z: np.random.randn(N, D),
    }
    out = sess.run([c, grad_x, grad_y, grad_z],
                    feed_dict=values)
    c_val, grad_x_val, grad_y_val, grad_z_val = out
```

PyTorch

```
import torch
from torch.autograd import Variable

N, D = 3, 4

x = Variable(torch.randn(N, D).cuda(),
              requires_grad=True)
y = Variable(torch.randn(N, D).cuda(),
              requires_grad=True)
z = Variable(torch.randn(N, D).cuda(),
              requires_grad=True)

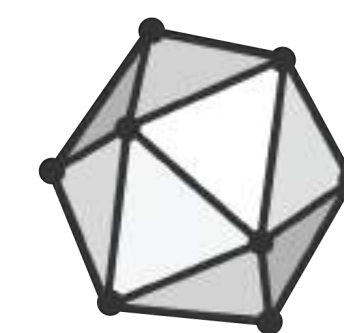
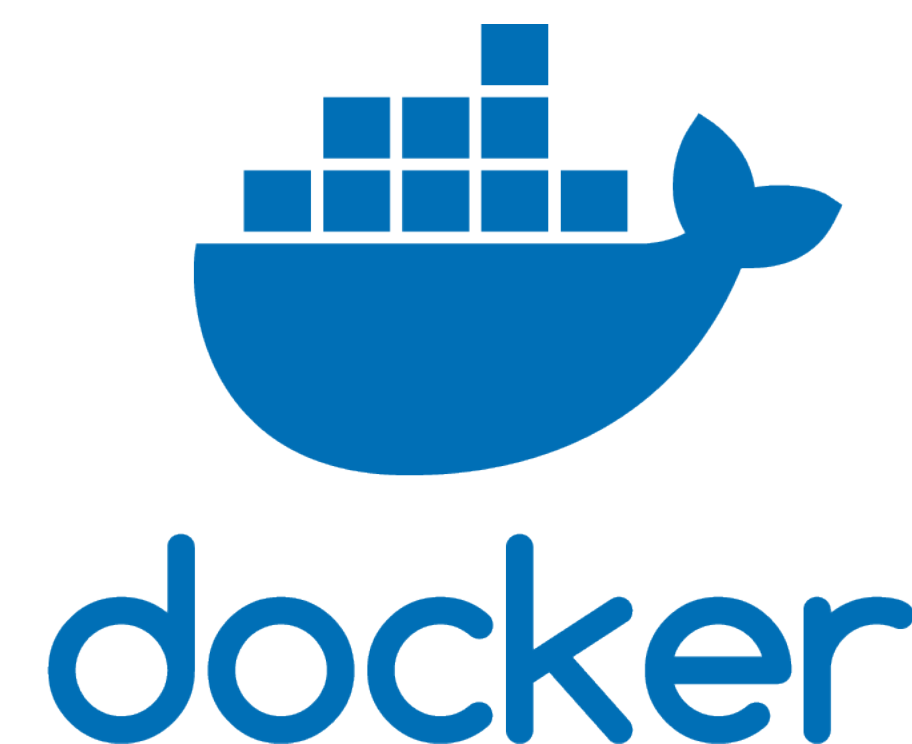
a = x * y
b = a + z
c = torch.sum(b)

c.backward()

print(x.grad.data)
print(y.grad.data)
print(z.grad.data)
```


部署运维

- 容错：检查点机制
- 数据预处理：Hadoop & Spark
- Docker & Kubernetes
- 框架转换：ONNX



ONNX

kubernetes

TABLE OF CONTENTES

深度学习框架的定位

深度学习框架的最佳实践

深度学习框架当前技术焦点

主流深度学习框架点评

展望

Define-and-run 与 Define-by-run

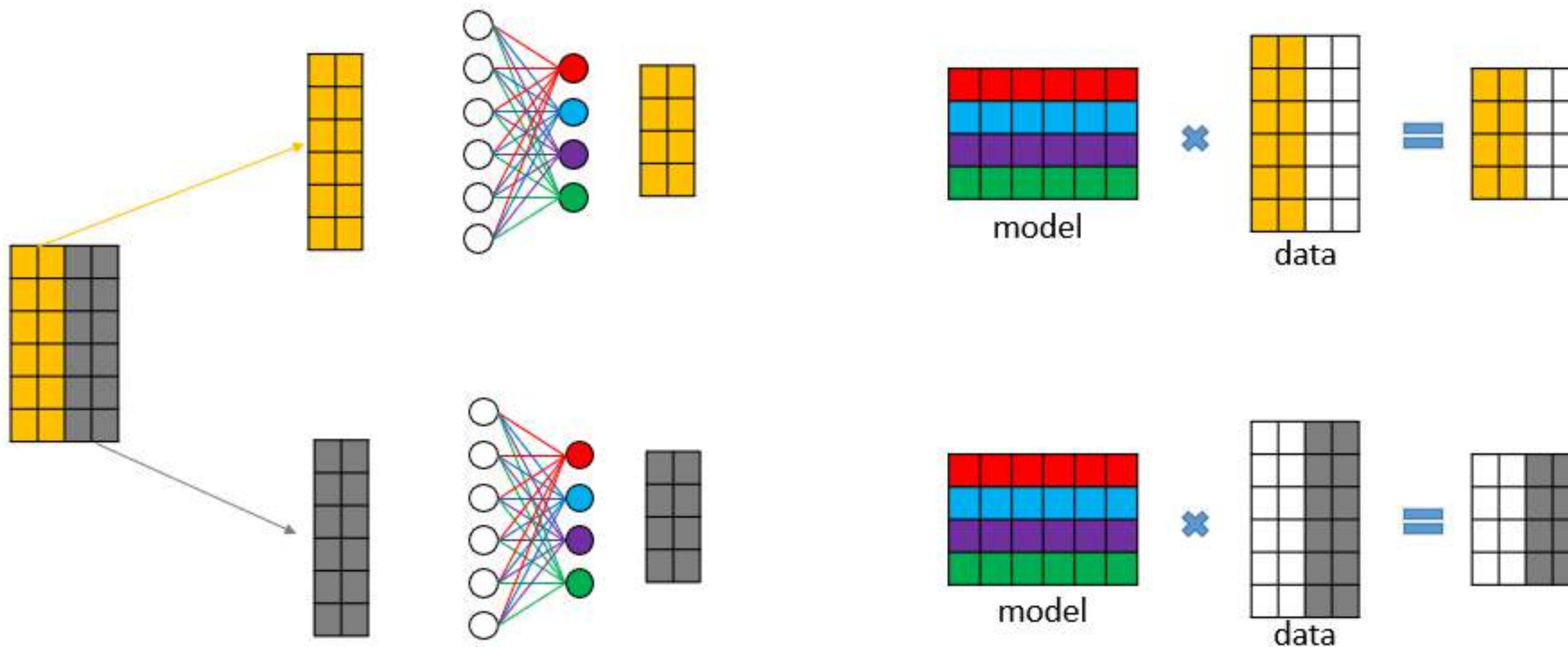
- Lazy evaluation 与 eager evaluation
- Declarative programming 与 imperative programming
- Dataflow 与 control flow
- 高效性与灵活性

分布式训练

- 数据并行已经解决
- 模型并行支持的不好
- 流水线并行支持的不好
- 参数服务器，或者MPI，或者Client-Master-Worker架构

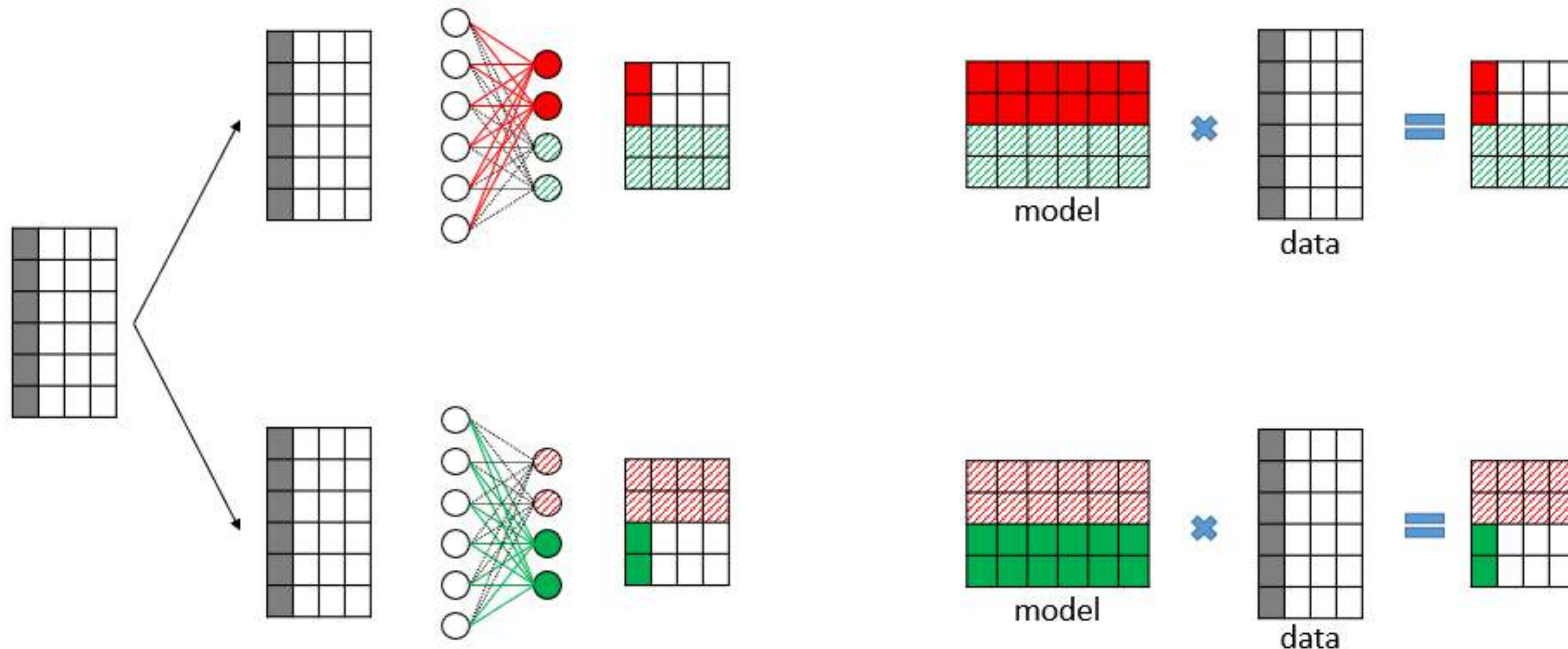
数据并行

- 所有开源深度学习框架都支持



模型并行

- 极少开源框架支持



流水线并行

- 磁盘IO, 网络, PCIe, 计算

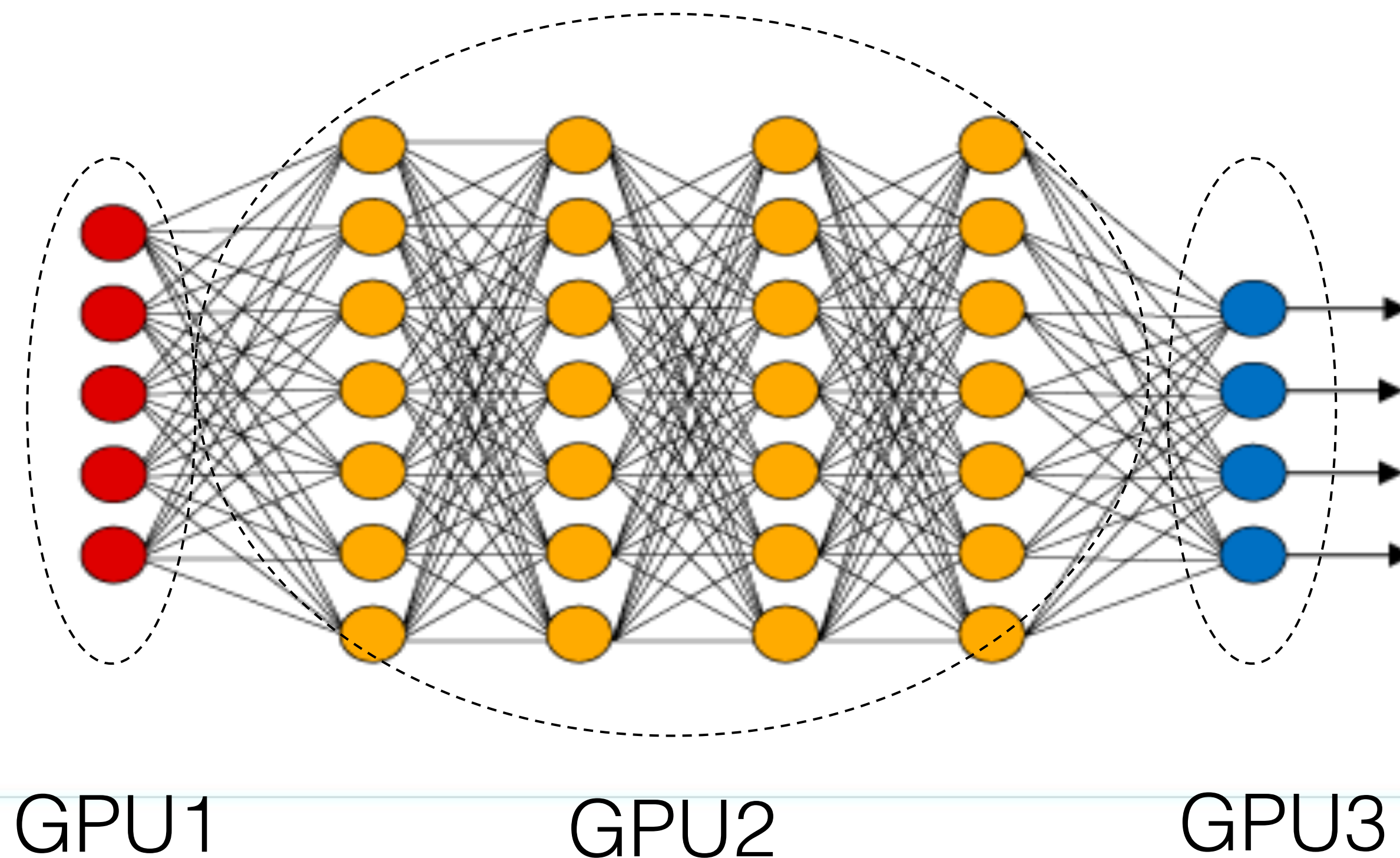


TABLE OF CONTENTES

深度学习框架的定位

深度学习框架的最佳实践

深度学习框架当前技术焦点

主流深度学习框架点评

展望

我比较关注的框架





TensorFlow

系统完整度最高

需要解决性能问题（须壮士断腕）

训练大规模RNN/LSTM的唯一选择

PYTORCH

NLP 应用首选
单机场景下的王者
难以支持大规模应用



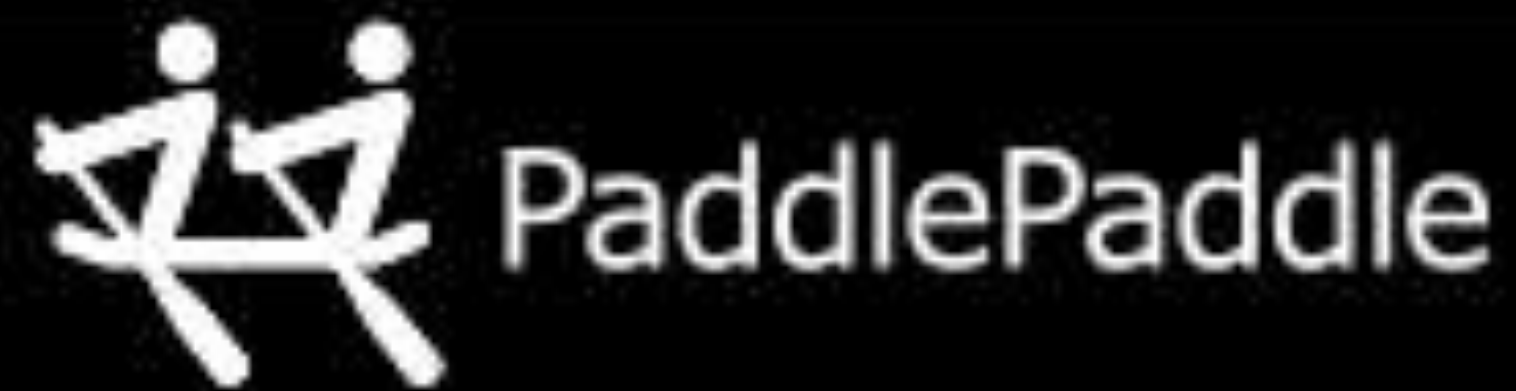
大胆尝试各种新技术（杂）

CV场景有优势

TVM 是一个很有特色微观优化模块



CV 场景有优势
代码干净利落
网络库gloo独具特色



百度实战认证
上一版架构比较老
重构版优势还不明朗

开发一个深度学习框架不难
开发一个显著超越前人的系统很难

TABLE OF CONTENTES

深度学习框架的定位

深度学习框架的最佳实践

深度学习框架当前技术焦点

主流深度学习框架点评

展望

展望

- 计算机视觉领域也需要模型并行（不仅数据大，模型也大）
- 模型并行得到解决
- 深度学习向更多场景渗透
- 深度学习框架技术逐渐收敛（同质化严重，需要新的空气）
- 深度学习框架变得像Hadoop一样不可或缺

Thanks!

