

框架之争



ZOMI 酱

Building a better connected world



www.hiascend.com
www.mindspore.cn

关于本内容

1. 内容背景

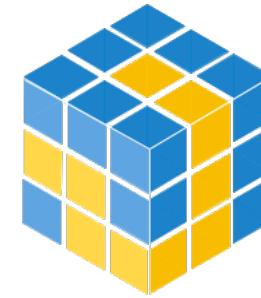
- AI框架的基础介绍

2. 具体内容

- AI框架作用：深度学习基础 – AI框架的作用 - AI框架的目的
- AI框架之争：第一代框架 – 第二代框架 – 第三代框架
- 编程范式：声明式编程 - 命令式编程

Gen 1 pre-2010

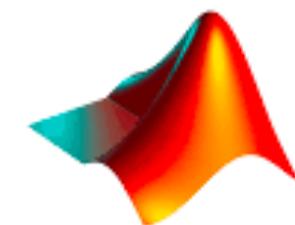
- 解决问题
 - 机器学习ML中缺乏算法库
 - 稳定和统一的神经网络NN定义
- 主要特点 - Library
 - 脚本式编程
 - 通过简单配置的形式定义神经网络
 - 针对特殊的ML、NN算法提供接口（MATLAB、SciPy）
 - 针对矩阵计算提供特定的计算接口（NumPy）
- 优点
 - 提供了一定程度的可编程性
 - 计算性能高：支持CPU加速计算



NumPy



SciPy



MATLAB®

Gen 1 pre-2010

解决问题

- 以CNN网络模型为主，由常用的layers组成，如：
Convolution, Pooling, BatchNorm, Activation等

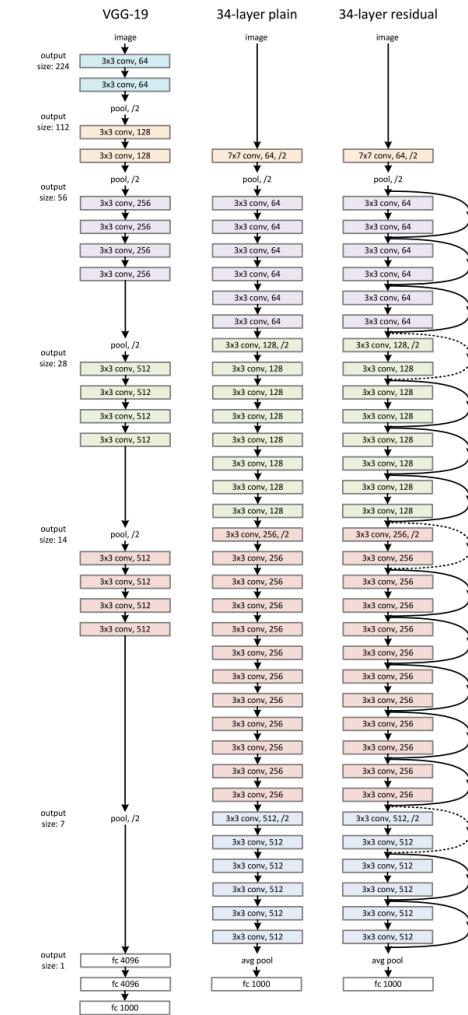
主要特点 – Layer Base

- 通过简单配置文件的形式定义神经网络
- 模型可由一些常用layer构成一个简单的图
- 框架提供每一个layer及其梯度计算实现
- 支持多设备加速：CPU和GPU的高效计算

优点

- 提供了一定程度的可编程性
- 计算性能高：支持GPU加速计算

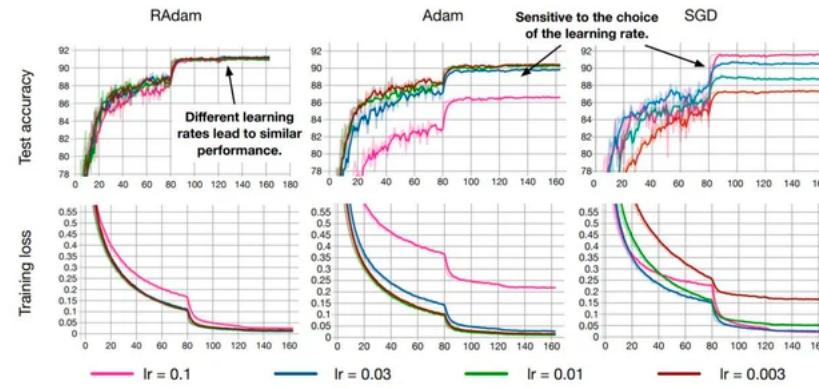
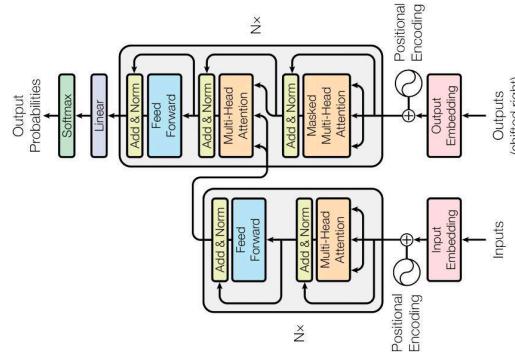
```
layer {
    name: "conv2"
    type: "Convolution"
    bottom: "pool1"
    top: "conv2"
    param {
        lr_mult: 1
    }
    param {
        lr_mult: 2
    }
    convolution_param {
        num_output: 50
        kernel_size: 5
        stride: 1
        weight_filler {
            type: "xavier"
        }
        bias_filler {
            type: "constant"
        }
    }
}
layer {
    name: "pool2"
    type: "Pooling"
    bottom: "conv2"
    top: "pool2"
    pooling_param {
        pool: MAX
        kernel_size: 2
        stride: 2
    }
}
```



Gen 1 pre-2010 limitation (I)

灵活性的限制难以满足深度学习的快速发展

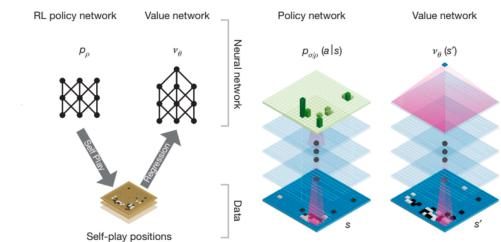
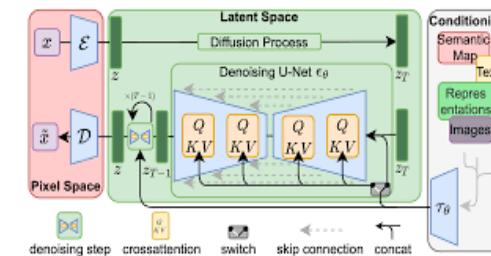
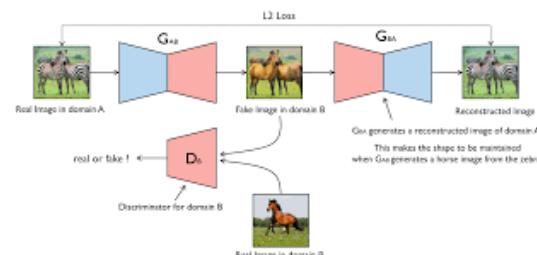
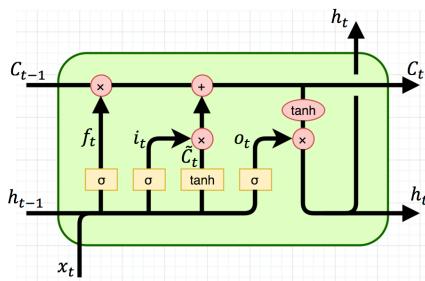
- 层出不穷的新型网络结构
- 新层需要重新实现前向和后向计算
- 非高级语言实现，修改和定制化成本高
- 新优化器要求对梯度和参数进行更通用复杂的运算



Gen 1 pre-2010 limitation (II)

基于简单的“前向+后向”的训练模式难以满足新的训练模式

- 循环神经网络需要引入控制流
- 对抗神经网络需要两个网络交替训练
- 强化学习模型需要和外部环境进行交互

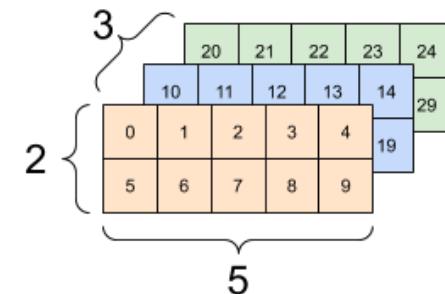
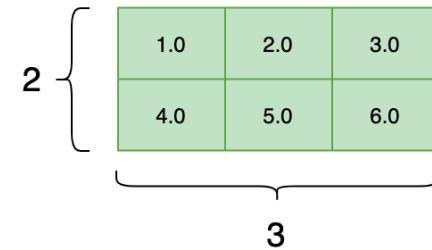


Gen 2 Present : Base DAG

基于数据流图（DAG）的计算框架

基本数据结构 : Tensor 张量

- Tensor形状 : [2, 3, 4, 5]
- 元素类型 : int, float, string, etc.



基本运算单元 : Operator 算子

- 由最基本的代数算子组成
- 根据深度学习结构组成复杂算子
- N个输入Tensor , M个输出Tensor

Add	Log	While
Sub	MatMul	Merge
Mul	Conv	BroadCast
Div	BatchNorm	Reduce
Relu	Loss	Map
Floor	Sigmoid

Gen 2 Present : Base DAG

基于数据流图(DAG)的计算框架

DAG 表示计算逻辑和状态

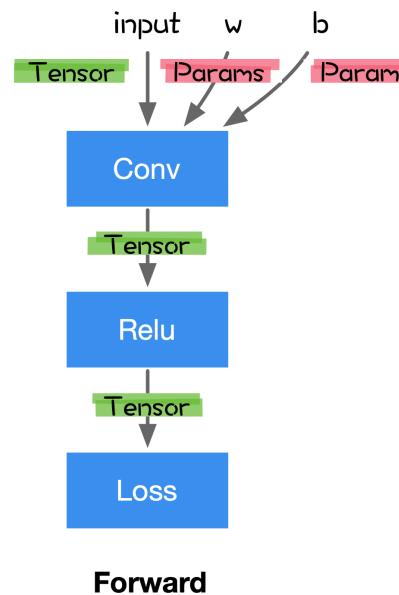
- 节点代表 Operator
- 边代表 Tensor

特殊的操作

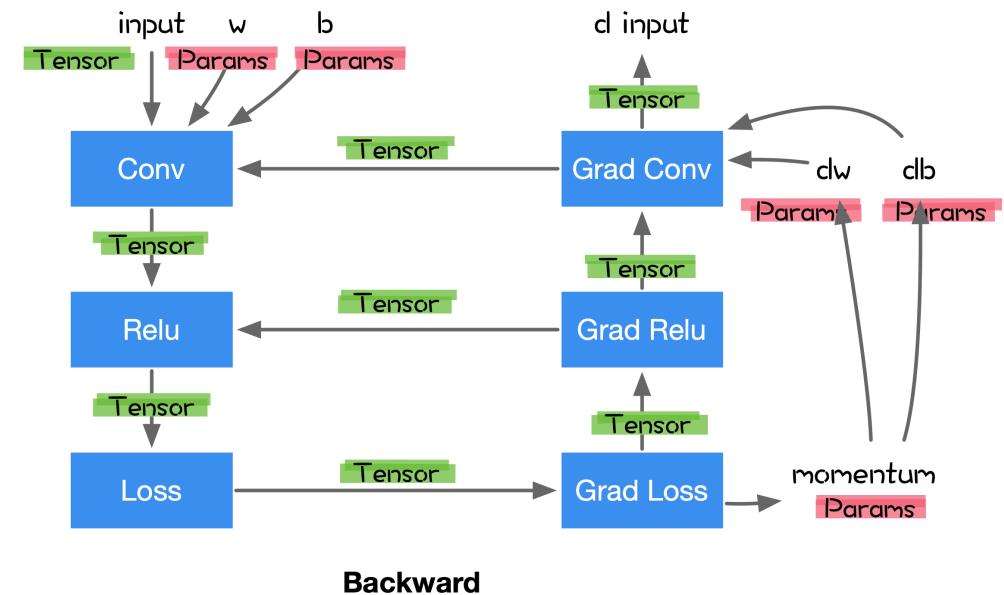
- 如：For/While 等构建控制流

特殊的边

- 如：控制边表示节点间依赖



Forward

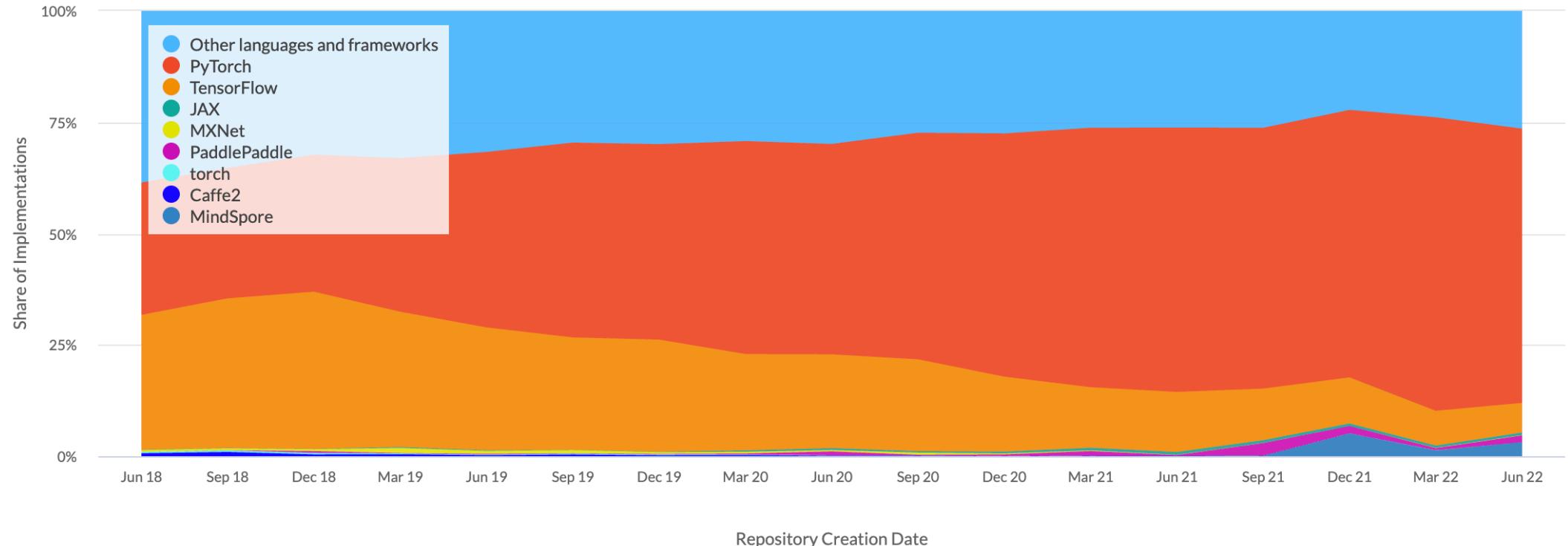


Backward

Gen 2 Present : Base DAG

TensorFlow vs PyTorch 代表深度学习框架两种不同的设计路径：
系统性能优先改善灵活性和灵活性易用性优先改善系统性能

Paper Implementations grouped by framework



Gen 3 Present : toward DSL

设计特定领域语言(Domain-Specific Language , DSL)



Gen 3 Present : toward DSL

兼顾编程的灵活性和计算的高效性

- 提高描述神经网络算法表达能力和编程灵活性
- 通过编译期优化技术来改善运行时性能



前端编程语言和接口

Python, Lua, R, C++

自动求导 (Auto Differentiation)



统一模型表示

计算流图 DAG

图的优化与调度执行

Batching, Cache, Overlap



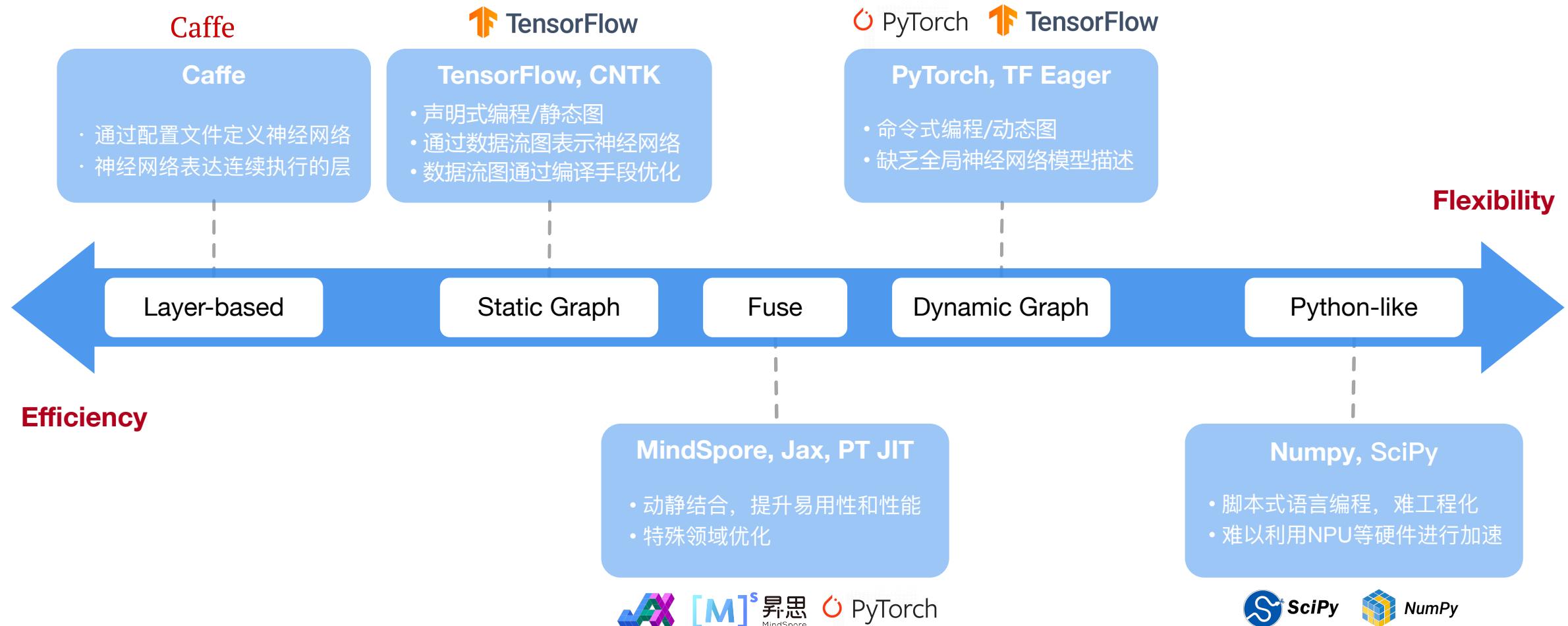
内核代码优化与编译

GPU kernel, auto kernel generation

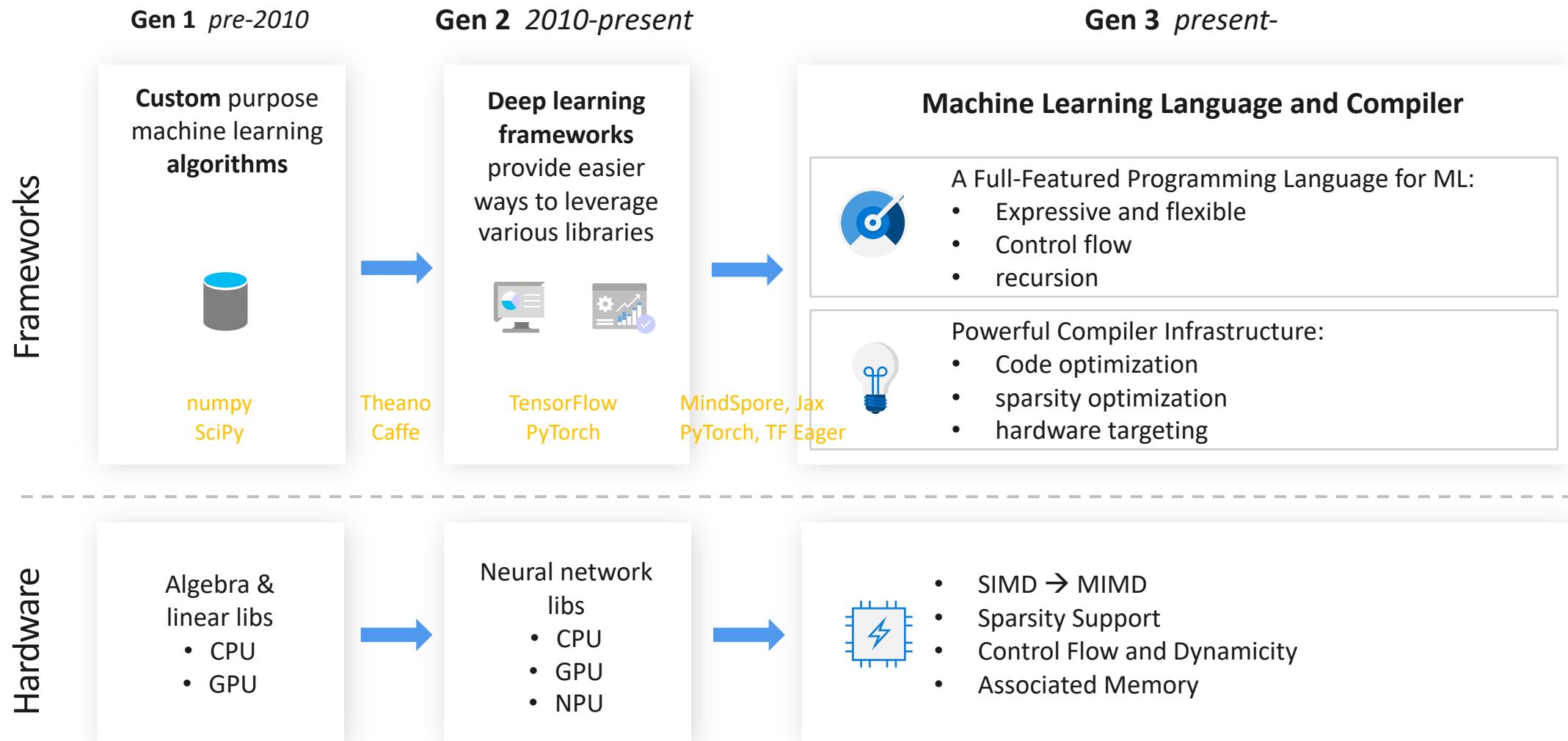
计算硬件

CPU, GPU, RDMA devices

Advances in AI frameworks



Advances in AI frameworks



Summary

1. 回顾了AI框架的发展趋势
 - 第一代解决深度学习编程问题
 - 第二代AI框架加速科研和产业落地
 - 第三代结合特定领域语言和任务，快速发展
2. 一起学习了AI框架随着的软硬件的发展升级而共同发展



BUILDING A BETTER CONNECTED WORLD

THANK YOU

Copyright©2014 Huawei Technologies Co., Ltd. All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.