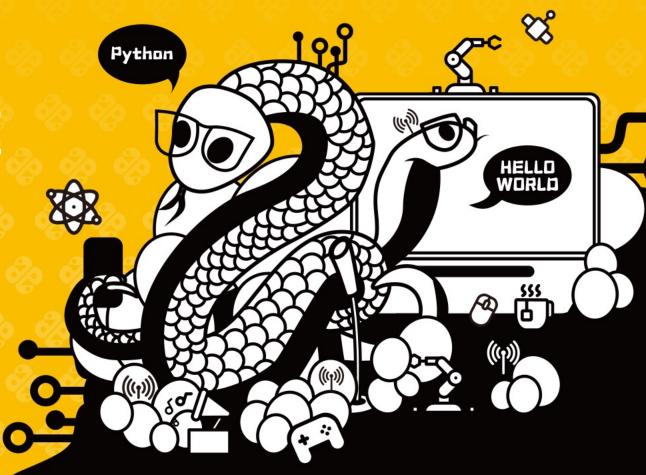


Python for Good

PyCon China 2022

基于Python的深度学习框 架设计与实现

主讲人: 刘凡平



Python for Good

>>>> PyCon China 2022

- 一、背景
- 二、原理:深度学习框架的一般性结构
- 三、设计
- 四、应用案例
- 五、思考

Python for Good>>>> PyCon China 2022

深度学习框架是包含深度学习模型设计、训练和验证的一套标准接口、特性库和工具包,集成深度学习的算法封装、数据调用以及计算资源的使用,同时面向开发者提供了开发界面和高效的执行平台,是算法工程师的必备工具之一。









theano

PYTORCH

美国互联网对AI底层技术战略性投入力度较大,但中国的AI产业主要受需求拉动,大多数人工智能公司布局应用层。

一、背景:深度学习框架的演进

Python for Good>>>> PyCon China 2022

深度学习框架的演进

萌芽阶段

时间范围:2000年-2012年左右

使用特点

1、API复杂;

2、手写神经网络结构;

- 3、无GPU算力;

应用情况:受限于算力不足,神经网络技术影响力相对有限,更多以机器学习工具的形式产生。

示例框架: OpenNN、Torch

发展阶段

时间范围:2012年左右-2016年左右

使用特点

1、生态友好;

2、支持GPU以及分布式计算;

3、性能优化、支持扩展等;

应用情况:使用效果高于之前传统机器学习水平,融合CNN、RNN等经典

模型结构

示例框架: Caffe、Chainer、Theano等

深化阶段

时间范围: 2020年-至今

使用特点

1、全场景支持;

- 2、超大规模AI;

3、安全可信等;

应用情况:随着人工智能的进一步发展,新的趋势不断涌现。要求 AI 框架最大化的实现编译优化,更好地利用算力、调动算力,充分发挥硬件资源的潜力。

示例: 华为推出昇思 MindSpore、旷视推出天元 MegEngine等

稳定阶段

时间范围:2016年左右-2020年

使用特点

1、编译层优化;

- 2、多场景、多任务支持;

3、丰富的开发套件;

应用情况:在计算机视觉和自然语言处理的应用效果得到极大的提升,经过

激烈的竞争后,最终形成了TensorFlow 和PyTorch两大阵营

示例框架:PyTorch、TensorFlow、MXNet、PaddlePaddle等

一、背景:深度学习框架的重要性

Python for Good>>>> PyCon China 2022

对下能够屏蔽底层差异并提供良好的执行性能,对上支撑 AI 应用算法模型搭建, 提供算法工程化实现的标准环境。 1、是整个人工智能技术体系的核心 承担着技术生态中基础环节,是学术创新与产业商业化的重要载体,助力人工智能由理 论走入实践,快速进入场景化应用。 2、是算法落地的基础技术工具 深度学习框架重要性 智能经济时代的硬件计算已经显示出从串行迁移到并行计算的趋势。AI框架能够促进生 态圈关联及外围的芯片、系统、软硬件平台等产业发展,从而促进人工智能核心生态圈 的建设。 3、是未来智能经济时代的操作系统

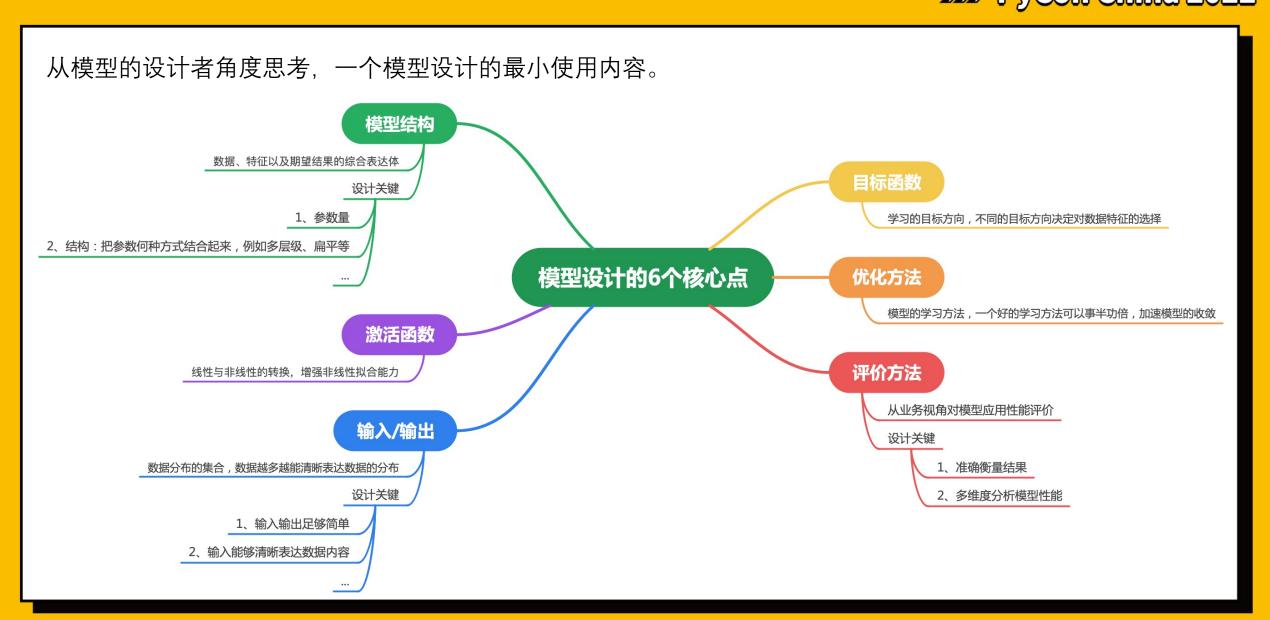
二、原理:深度学习框架结构

Python for Good>>>> PyCon China 2022

社区 面向应用服务提供解 经典套件库 经典模型库 AI+科学计算 分词/图像分类/... Bert/GNN/... 生物制药/物理/... 层 文档 决方案或一般性方法。 并行及优化组件 科学计算组件 安全可信组件 工具组件 组 件 提供模型生命周期中 自动并行控制 模型可解释性 训练可视化 数值计算 科配置的各类功能组件。 高阶优化器 逻辑计算 调试器 程序 标准API接口 不同语言SDK 实现框架最基础、最 内存优化 模型轻量化 图笪融合 分布式并行 核心的功能,帮助开发者 计算图标识方法 自动微分 动静转化 屏蔽底层硬件技术细节。 计算 通信协议 CPU/GPU

三、设计:最小化的深度学习框架

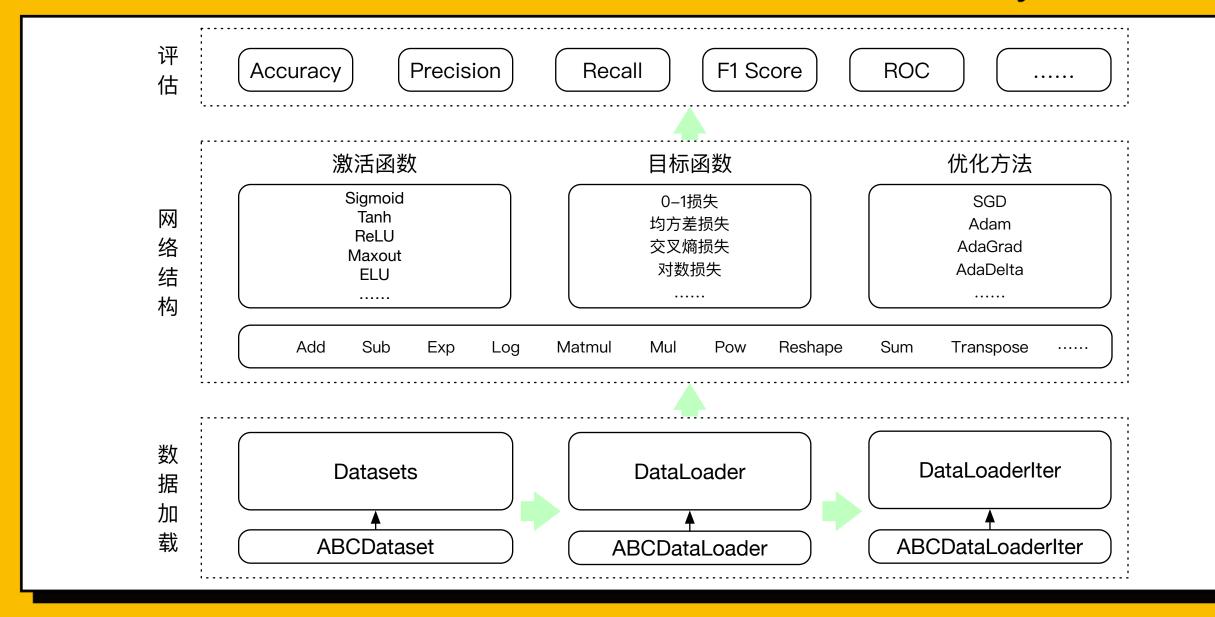
Python for Good>>>> PyCon China 2022



三、设计:最小MVP深度学习框架的层次逻辑

Python for Good

>>>> PyCon China 2022



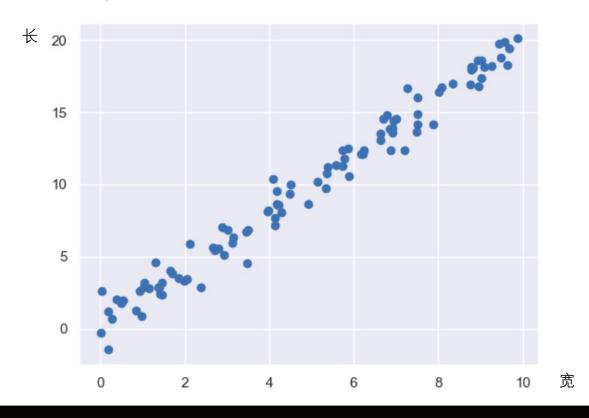
四、应用:简单回归问题的实现(一)

Python for Good>>>> PyCon China 2022

1、定义问题

已知某类工业零件不同长与宽的产品数据集,求长与宽的关系。

2、宽与高的数据关系示意图:



四、应用:简单回归问题的实现(二)

Python for Good

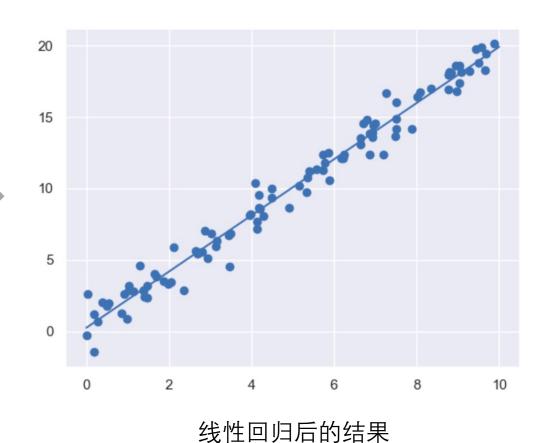
>>>> PyCon China 2022

模型结构:

class SampleModel(nn.Module):

```
def __init__(self):
  super(SampleModel, self).__init__()
  self.line_layer = nn.Tensor.line(1, 1)
```

def forward(self, x: nn.Tensor) -> nn.Tensor:
out = self.line_layer(x)
return out



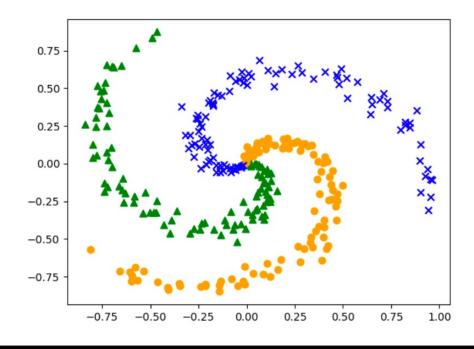
四、应用:简单分类问题的实现(一)

Python for Good>>>> PyCon China 2022

1、定义问题

100个二维数据点,一共分为3个类别,分别识别出他们的分类信息。

2、示例图



左侧为测试数据的样例(训练数据的分布与此类似),目前已经通过三种数据颜色和标识标注了他们所属的类别,要求通过模型分辨出划分出他们的类别。

四、应用:简单分类问题的实现(二)

Python for Good

>>>> PyCon China 2022

模型结构:

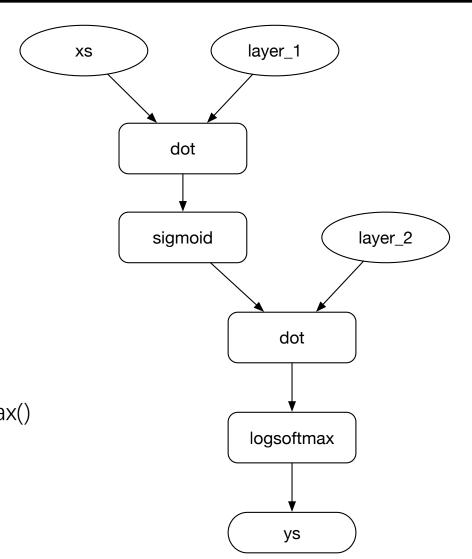
class SampleModel(nn.Module):

```
def __init__(self, in_size, hidden_size, out_size):
  super().__init__()
  self.layer_1 = nn.Tensor.uniform(in_size, hidden_size)
  self.layer_2 = nn.Tensor.uniform(hidden_size, out_size)
```

def forward(self, xs):

#xs为原始的批量数据集合。

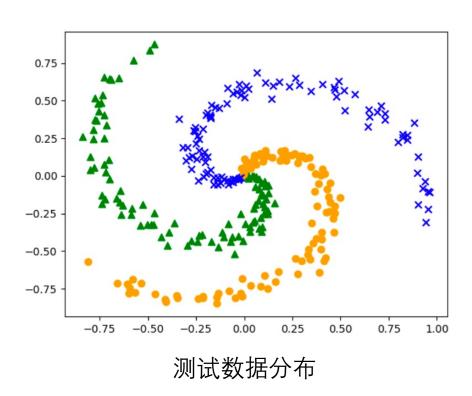
return xs.dot(self.layer_1).sigmoid().dot(self.layer_2).logsoftmax()

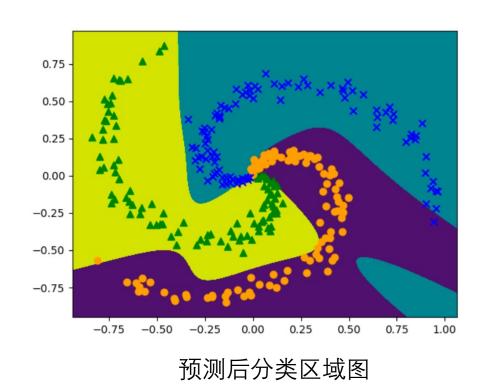


Python for Good

>>>> PyCon China 2022

训练10个epoch后的效果对比图:





Python for Good>>>>> PyCon China 2022

- 为什么要设计一个深度学习框架? 切勿以造轮子的初衷去设计深度学习框架,一切均需**围绕业务**进行。脱离业务的技术体系价值不大。
- 是否存在完美的深度学习框架?
- 一切以<mark>落地场景</mark>为根基,满足业务使用即可,不要过度设计,过度设计将会导致框架越来越复杂、 臃肿。

- 实现的深度学习框架与目前主流开源的结果计算结果不一致怎么办?
- 一方面深入原理,在掌握**系统理论基础**后去实现;另一方面每一个算法的实现细节都会不同,注意 实现**细节**。

Python for Good>>>>> PyCon China 2022





