



19.1 PyTorch简介

■ PyTorch

- 由Facebook开发的**开源神经网络框架**
- **动态**计算图，可根据计算需要实时改变计算图
- 对多维数组进行操作
- 可快速、简洁实现机器学习和深度学习

■ PyTorch特点

- **简洁**——设计追求最少的封装，避免重复造轮子
- **快速**——执行速度快
- **易用**——设计更符合人类思维，无需过多考虑框架本身约束



■ PyTorch与TensorFlow的差异

□ 可视化工具

TensorFlow 内置可视化工具**Tensorboard**，可展示模型图形、绘制标量变量、可视化分布等
PyTorch暂无独立可视化工具，但Tensorboard 也可用于PyTorch

□ 计算图

TensorFlow1.X： 静态计算图，

TensorFlow2.X： 动态计算图， 静态计算图

PyTorch： 仅支持动态计算图

计算图的正向传播是**立即执行的**，计算图在反向传播后**立即销毁**



■ PyTorch与TensorFlow的差异

□ 自动求导机制的差异

Pytorch 自动求导	TensorFlow 求导
计算梯度的两种方法 torch.autograd.backward() torch.autograd.grad()	创建 Variable 可训练变量, with GradientTape() as tape: 函数表达式 grad=tape.gradient(函数,自变量)

PyTorch: 通过反向传播 **backward** 方法实现求梯度计算, 求得的梯度将存在对应自变量张量的**grad属性**中

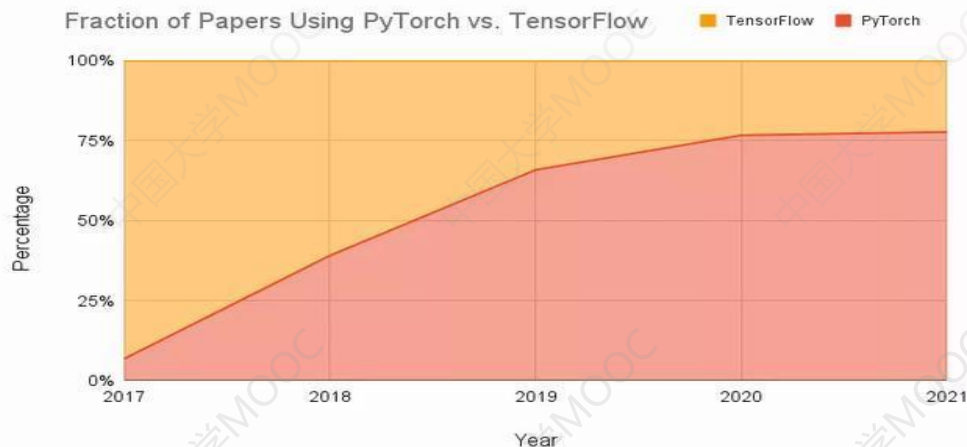
TensorFlow: 使用梯度磁带**tf.GradientTape**记录正向运算过程, 然后反播磁带自动得到梯度值



■ PyTorch VS TensorFlow

□ 模型可用性

两者均有成熟的官方模型存储库，但在学术研究领域，PyTorch应用越来越普遍，在近几年在学术论文中的使用率从大约 7% 增长到近80%



□ 模型部署

TensorFlow 仍然是面向部署的应用程序的首选框架，TensorFlow Serving和TensorFlow Lite可让用户轻松地在云、服务器、移动设备和 IoT 设备上部署。各大公司在招聘深度学习工程师时，大部分都要求掌握TensorFlow框架。

PyTorch 近年来接连推出TorchServe和PyTorch Live提供了本地部署工具，但仍然只针对移动端，并且许多部署工具还处于起步阶段。



□ 生态系统

PyTorch和TensorFlow的相关生态都做得很好。如PyTorch中的TorchX是一个用于快速构建和部署机器学习应用程序的 SDK，TorchX 包括 Training Session Manager API，可将分布式 PyTorch 应用程序启动到受支持的调度程序上；TensorFlow Hub是一个经过训练的机器学习模型库可以进行微调，用户只需几行代码就可以使用像 BERT 这样的语言模型等等。



□ 对比总结

近年来，PyTorch和TensorFlow都在各方面不断完善，两者各有特色，无分好坏，可根据用户当下的**项目技术需要**或**研究需求**进行选择。



□ TensorFlow

强大而成熟的深度学习库，有强大的可视化性能，以及用于高水平模型开发和部署的多个选项。如果你符合以下情况，TensorFlow对你而言会是个很好的选择：

- 开发用于生产的模型
- 开发需要在移动平台上部署的模型
- 想要非常好的社区支持和较为全面的帮助文档
- 想要丰富的多种形式的学习资源
- 想要或需要使用 Tensorboard
- 需要用到大规模分布式模型训练



□ PyTorch

PyTorch虽发展迅速，但仍然是个比较年轻的框架。如果符合以下情况，PyTorch对你而言会是个很好的选择：

- 正在做机器学习、深度学习的**学术研究**
- 开发的产品在**非功能性需求**方面要求不高
- 想要获得**更好的开发和调试经验**
- **喜欢**更具**Python形式**的编程风格

