

Basic_00_00_为什么选择pytorch

2019年12月26日 10:00

① • **简洁**: PyTorch 的设计追求最少的封装, 尽量避免重复造轮子。不像 TensorFlow 中充斥着 session、graph、operation、name_scope、variable、tensor、layer 等全新的概念, PyTorch 的设计遵循 `tensor→variable(autograd)→nn.Module` 三个由低到高的抽象层次, 分别代表高维数组(张量)、自动求导(变量)和神经网络(层/模块), 而且这三个抽象之间联系紧密, 可以同时进行修改和操作。

简洁的设计带来的另外一个好处就是代码易于理解。PyTorch 的源码只有 TensorFlow 的十分之一左右, 更少的抽象、更直观的设计使得 PyTorch 的源码十分易于阅读。在笔者眼里, PyTorch 的源码甚至比许多框架的文档更容易理解。

② • **速度**: PyTorch 的灵活性不以速度为代价, 在许多评测中, PyTorch 的速度表现胜过 TensorFlow 和 Keras 等框架^{①,②}。框架的运行速度和程序员的编码水平有极大关系, 但同样的算法, 使用 PyTorch 实现的那个更有可能快过用其他框架实现的。

③ • **易用**: PyTorch 是所有的框架中面向对象设计的最优雅的一个。PyTorch 的面向对象的接口设计来源于 Torch, 而 Torch 的接口设计以灵活易用而著称, Keras 作者最初就是受 Torch 的启发才开发了 Keras。PyTorch 继承了 Torch 的衣钵, 尤其是 API 的设计和模块的接口都与 Torch 高度一致。PyTorch 的设计最符合人们的思维, 它让用户尽可能地专注于实现自己的想法, 即所思即所得, 不需要考虑太多关于框架本身的束缚。

- **活跃的社区**: PyTorch 提供了完整的文档, 循序渐进的指南, 作者亲自维护的论坛^③供用户交流和请教问题。Facebook 人工智能研究院对 PyTorch 提供了强力支持, 作为当今排名前三的深度学习研究机构, FAIR 的支持足以确保 PyTorch 获得持续的开发更新, 不至于像许多由个人开发的框架那样昙花一现。

在 PyTorch 推出不到一年的时间内, 各类深度学习问题都有利用 PyTorch 实现的解决方案在 GitHub 上开源。同时也有许多新发表的论文采用 PyTorch 作为论文实现的工具, PyTorch 正在受到越来越多人的追捧^④。