

# 一、前言

本分析报告的作者是中国科学院大学 2019 级本科生夏瑞阳，本报告是作者在《面向对象程序设计》课程中选取 PyTorch 的 nn 模块进行分析所完成的作业，因此本报告将不会把篇幅过多放在 PyTorch 具体源码实现的解析与算法设计的讨论，而将更侧重于主要功能流程设计、类的设计与类间关系、面向对象的设计原则和设计模式分析等方面。

## 二、导言

导言中将简要介绍 PyTorch 与 PyTorch.nn 的主要功能、流程与模块。

### 1、什么是 PyTorch

PyTorch 是 Python 的一个深度学习库，是一个能够在 CPU 或 GPU 上运行并解决各类深度学习问题的深度学习框架。相较于其他深度学习框架，它具有以下几个优势：

#### ①简洁：

PyTorch 的原身是 Lua 语言上的 Torch，因此 PyTorch 继承了 Torch 的设计原则，追求最少的封装，尽量避免重复造轮子。与另一个热门的深度学习框架 TensorFlow 相比，PyTorch 中没有 session、graph、tensor、layer、name\_scope、operation 等抽象层次，而是只有 Tensor->Variable->Module 的由低到高的抽象层次。因此这使得 PyTorch 的源代码非常易读。

#### ②速度：

PyTorch 能提供强大的 GPU 加速和更高效的算法来支持网络中的计算，在许多评测中，PyTorch 的速度表现胜过 TensorFlow 和 Keras 等深度学习框架。

#### ③易用：

PyTorch 能基于磁带式的自动微分系统来构建一个深度神经网络。即设计者只需要自己重载 Module 类的 `__init__()` 和 `forward()` 方法，而网络反向传播的部分由 `torch.autograd` 模块自动完成，无需设计者再手动实现。此外，它也能与 NumPy、SciPy、Cython 等 Python 库很好地兼容，从而使得设计者能够复用他们以前的代码。

Pytorch 中除了 nn、autograd 等模块外，还有许多常用模块，现介绍如下：

名称	描述
torch	一个有着强大 GPU 支持的张量库，是 Pytorch 的基本组件
torch.autograd	一个磁带式的自动微分系统，能支持 torch 内所有张量可微操作
torch.jit	一个通过 Pytorch 源码生成可序列化和可优化模型的编译栈
torch.nn	一个与自动微分系统深度集成的神经网络库，十分灵活便捷
torch.multiprocessing	一个支持多进程的库，并且有着强大的张量间内存共享功能
torch.utils	一个工具库，其中有 DataLoader 等便利性的工具

## 2、PyTorch.nn 中所包含的主要类

pytorch.nn 中定义了许多个类，其中主要有以下几个：Module、Parameter、Functional、grad、init、utils 等。在之后的报告中会基于这些类对 pytorch.nn 进行详细分析。