

# PyTorch 极简入门路线

## 一、Python 学习路线

### 1.1 简介

Python 简单易学，有其他编程基础的同学能够快速上手。

关于版本：Python 2 官方已经放弃更新支持，所以建议学习和使用 Python3 的版本。

关于编辑器：推荐使用 PyCharm，通过学生邮箱注册可以免费使用，也可以通过品破解教程进行破解。

关于语法：学习 Python 基础语法，例如从基本的变量赋值、条件循环、到文件操作等。有其他面向对象语言基础的可以更容易上手 Python，Python 是一个完全面向对象的语言，在 Python 里一切皆是对象。

### 1.2 推荐学习路线

#### STEP 1: 基础教程

(1) 选择一，直接在线学习，在线运行（参考学习总时长：半天-1 天）

<https://www.runoob.com/python3/python3-tutorial.html>

(2) 选择二，或者选择廖雪峰的基础教程（参考学习总时长：半天-1 天）：

<https://www.liaoxuefeng.com/wiki/1016959663602400>

(3) 选择三，如果不喜欢文字版的，可以看莫烦的视频教程（视频看完即可，感觉已掌握和啰嗦之处可以快进和倍速）：

<https://morvanzhou.github.io/tutorials/python-basic/basic/>

#### STEP 2: 实例教程

Github 上的 100 天从新手到大师教程，可以看看里面的练习案例，有的很有趣，而且动手实现可以增加对知识的认知：

<https://github.com/jackfrued/Python-100-Days>

看完以上基础教程，就大概了解了 Python，其中 Python 里面重要的几个对象要掌握牢固，这是我们最常接触的：

list [ ]

dict { }

`tuple()`

然后看完教程只是保证你对 Python 有个第一印象，并不要求死记硬背教程，最重要的是在实践中慢慢掌握，在实践中遇到什么问题不懂，学会自己用搜索引擎解决，在这个渐进过程中提升自己。

## 二、PyTorch 学习路线

### 2.1 简介

了解完 Python，如果想入门机器学习或者深度学习，推荐使用框架 PyTorch。对于我们做学术而言，能够用其快速实现自己的想法，快速、容易地进行调试。因为 PyTorch 是动态图，完全按照 Python 哲学和语法设计的，所以更适合我们上手。而 TensorFlow 的优势是生产环境，但其为静态图，复杂的 API 设计等对于初学者并不友好。

目前学术圈的最新论文几乎都倾向于开源基于 PyTorch 的代码，所以能够熟练使用这一框架，是必备技能。

关于版本：PyTorch 建议学习 1.0 以上版本，最好能够对最新发布版本进行学习。

### 2.2 推荐学习路线

#### STEP 1: 基础教程

最好的 PyTorch 教程是官方的，没有其他教程比官方教程更适合入门了。Deep Learning with PyTorch: A 60 Minute Blitz 里面有 5 个小教程（参考学习总时长：1 小时-2 小时），静下心来认真看，看完后可以对 PyTorch 有个初步的了解。

#### STEP 2: 进阶教程

在 <https://pytorch.org/tutorials/> 下面接着看自己感兴趣的或者和自己方向相关的教程。可以花一天甚至一周的时间掌握数据模型是如何定义，训练的。注意一定要亲自动手写代码，不要粘贴复制，并亲自跑完一个完整的程序（比如 MINIST 数据分类）。看完之后，要理解掌握以下概念：

1. 模型是什么: `torch.nn.Module`
2. 如何加载数据: `torch.utils.data.DataLoader`
3. 有了模型，数据，如何训练: `backward`, `optimizer`, `schduler`
4. 前向传播 `forward` 与反向传播 `backward` 的区别

5. 下面代码的含义是什么，为什么要这么做

```
optimizer.zero_grad()
```

```
loss.backward()
```

```
optimizer.step()
```

最后，遇到不懂的函数，通过 <https://pytorch.org/docs/stable/> 查看官方文档，这是最好的解释。

同时如果时间富余，也可以多看看 PyTorch 的论坛，里面有官方开发者对问题的亲自回复，很多问答质量都很高，这也是我觉得 PyTorch 远比其他框架好的地方：<https://discuss.pytorch.org/>。

Github 上有人总结了关于 PyTorch 各个应用方向实现的代码：<https://github.com/bharathgs/Awesome-pytorch-list>，大家一定要多看代码，学习别人优雅的实现方式，先从模仿开始，才会有自己的创新。

2019 年 12 月 30 日

北京