



链滴

ApacheCN 深度学习译文集 20210125 更新

作者: [wizardforcel](#)

原文链接: <https://ld246.com/article/1611755230692>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

新增了七个教程：

- [PyTorch 中文官方教程 1.7](#)
 - [学习 PyTorch](#)
 - [PyTorch 深度学习：60 分钟的突击](#)
 - [张量](#)
 - [torch.autograd 的简要介绍](#)
 - [神经网络](#)
 - [训练分类器](#)
 - [通过示例学习 PyTorch](#)
 - [热身：NumPy](#)
 - [PyTorch：张量](#)
 - [PyTorch：张量和 Autograd](#)
 - [PyTorch：定义新的 Autograd 函数](#)
 - [PyTorch：nn](#)
 - [PyTorch：optim](#)
 - [PyTorch：自定义 nn 模块](#)
 - [PyTorch：控制流 + 权重共享](#)
 - [torch.nn 到底是什么？](#)
 - [使用 TensorBoard 可视化模型，数据和训练](#)
 - [图片/视频](#)
 - [torchvision 对象检测微调教程](#)
 - [计算机视觉的迁移学习教程](#)
 - [对抗示例生成](#)
 - [DCGAN 教程](#)
 - [音频](#)
 - [音频 I/O 和 torchaudio 的预处理](#)
 - [使用 torchaudio 的语音命令识别](#)
 - [文本](#)
 - [使用 nn.Transformer 和 torchtext 的序列到序列建模](#)
 - [从零开始的 NLP：使用字符级 RNN 分类名称](#)
 - [从零开始的 NLP：使用字符级 RNN 生成名称](#)
 - [从零开始的 NLP：使用序列到序列网络和注意力的翻译](#)
 - [使用 torchtext 的文本分类](#)
 - [torchtext 语言翻译](#)

- 强化学习
 - 强化学习 (DQN) 教程
 - 训练玩马里奥的 RL 智能体
- 在生产中部署 PyTorch 模型
 - 通过使用 Flask 的 REST API 在 Python 中部署 PyTorch
 - TorchScript 简介
 - 在 C++ 中加载 TorchScript 模型
 - 将模型从 PyTorch 导出到 ONNX 并使用 ONNX 运行时运行它 (可选)
- 前端 API
 - PyTorch 中的命名张量简介 (原型)
 - PyTorch 中通道在最后的内存格式 (beta)
 - 使用 PyTorch C++ 前端
 - 自定义 C++ 和 CUDA 扩展
 - 使用自定义 C++ 运算符扩展 TorchScript
 - 使用自定义 C++ 类扩展 TorchScript
 - TorchScript 中的动态并行性
 - C++ 前端中的 Autograd
 - 在 C++ 中注册调度运算符
- 模型优化
 - 分析您的 PyTorch 模块
 - 使用 Ray Tune 的超参数调整
 - 模型剪裁教程
 - LSTM 单词语言模型上的动态量化 (beta)
 - BERT 上的动态量化 (Beta)
 - PyTorch 中使用 Eager 模式的静态量化 (beta)
 - 计算机视觉的量化迁移学习教程 (beta)
- 并行和分布式训练
 - PyTorch 分布式概述
 - 单机模型并行最佳实践
 - 分布式数据并行入门
 - 用 PyTorch 编写分布式应用
 - 分布式 RPC 框架入门
 - 使用分布式 RPC 框架实现参数服务器
 - 使用 RPC 的分布式管道并行化
 - 使用异步执行实现批量 RPC 处理

- 将分布式 [DataParallel](#) 与分布式 RPC 框架相结合
- [PyTorch 人工智能研讨会](#)
 - 零、前言
 - 一、深度学习和 PyTorch 简介
 - 二、神经网络的构建块
 - 三、使用 DNN 的分类问题
 - 四、卷积神经网络
 - 五、样式迁移
 - 六、使用 RNN 分析数据序列
 - 七、附录
- [Python 一次学习实用指南](#)
 - 零、前言
 - 第一部分：一次学习简介
 - 一、一次学习简介
 - 第二部分：深度学习架构
 - 二、基于指标的方法
 - 三、基于模型的方法
 - 四、基于优化的方法
 - 第三部分：其他方法和结论
 - 五、基于生成建模的方法
 - 六、总结和其他方法
- [Python 自然语言处理实用指南](#)
 - 零、前言
 - 第一部分：用于 NLP 的 PyTorch 1.x 的要点
 - 一、机器学习和深度学习的基础
 - 二、用于 NLP 的 PyTorch 1.x 入门
 - 第二部分：自然语言处理基础
 - 三、NLP 和文本嵌入
 - 四、文本预处理，词干提取和词形还原
 - 第三部分：使用 PyTorch 1.x 的实际 NLP 应用
 - 五、循环神经网络和情感分析
 - 六、用于文本分类的卷积神经网络
 - 七、使用序列到序列神经网络的文本翻译

- 八、使用基于注意力的神经网络构建聊天机器人
 - 九、前方的路
- PyTorch 人工智能基础知识
 - 零、前言
 - 一、使用 PyTorch 使用张量
 - 二、与神经网络协作
 - 三、用于计算机视觉的卷积神经网络
 - 四、用于 NLP 的循环神经网络
 - 五、迁移学习和 TensorBoard
 - 六、探索生成对抗网络
 - 七、深度强化学习
 - 八、在 PyTorch 中生产 AI 模型
- PyTorch 深度学习实用指南
 - 零、前言
 - 一、深度学习演练和 PyTorch 简介
 - 二、简单的神经网络
 - 三、深度学习工作流程
 - 四、计算机视觉
 - 五、序列数据处理
 - 六、生成网络
 - 七、强化学习
 - 八、生产中的 PyTorch
- TensorFlow 强化学习
 - 零、前言
 - 一、深度学习-架构和框架
 - 二、使用 OpenAI Gym 训练强化学习智能体
 - 三、马尔可夫决策过程
 - 四、策略梯度
 - 五、Q 学习和深度 Q 网络
 - 六、异步方法
 - 七、一切都是机器人-真正的战略游戏
 - 八、AlphaGo -最好的强化学习
 - 九、自动驾驶中的强化学习
 - 十、金融投资组合管理
 - 十一、机器人技术中的强化学习

- [十二、广告技术中的深度强化学习](#)
 - [十三、图像处理中的强化学习](#)
 - [十四、NLP 中的深度强化学习](#)
 - [十五、强化学习的其他主题](#)

下载

Docker

```
docker pull apacheecn0/apacheecn-dl-zh
docker run -tid -p <port>:80 apacheecn0/apacheecn-dl-zh
# 访问 http://localhost:{port}
```

PYPI

```
pip install apacheecn-dl-zh
apacheecn-dl-zh <port>
# 访问 http://localhost:{port}
```

NPM

```
npm install -g apacheecn-dl-zh
apacheecn-dl-zh <port>
# 访问 http://localhost:{port}
```

贡献指南

本项目需要校对，欢迎大家提交 Pull Request。

请您勇敢地去翻译和改进翻译。虽然我们追求卓越，但我们并不要求您做到十全十美，因此请不要担心因为翻译上犯错——在大部分情况下，我们的服务器已经记录所有的翻译，因此您不必担心会因为您失误遭到无法挽回的破坏。（改编自维基百科）