

深度学习与智能驾驶实践工坊教程课纲

目录

- 掌握深度学习基础（Day 1 上午）
- 实操深度学习案例（Day 1 下午）
- 探索智能驾驶模型（Day 2 上午）
- 实操智能驾驶案例（Day 2 下午）

第 1 章 掌握深度学习基础（Day 1 上午）

引言：怎样守护我的职业未来？

问：AI会写代码、做算法，似乎无所不能，作为一名码农，我该怎么办？ 答：拥有机器学习的完整知识体系，AI辅助下掌握全栈开发技能，成为理解AI的同行者。

1.1 WHAT：深度学习是什么？

1.1.1 深度学习与经典机器学习的联系与区别

- 深度学习与经典机器学习的核心区别在于“特征提取方式”：

维度	经典机器学习（如 SVM、决策树）	深度学习
特征提取	需要人工设计特征（如 HOG、SIFT）	自动学习特征表示（从原始数据到高层语义）
学习内容	模型只学习“如何组合已有特征”	端到端训练，减少人工设计环节
数据需求	适合小数据量	需要大量数据与计算资源
计算资源	计算需求较低	需要大量计算资源
可解释性	可解释性要求高的场景	可解释性相对较弱
适用场景	结构化数据、小样本场景	图像、语音、文本等高维（非结构化）数据上表现更优

- 深度学习与经典机器学习的联系是：
 - 都需要损失函数与优化器
 - 都面临过拟合与泛化问题
 - 深度学习可以视为“可学习特征”的机器学习扩展

1.1.2 深度学习发展历史

年份	里程碑	发明人/团队	说明
1986	反向传播算法	David Rumelhart, Geoffrey Hinton, Ronald Williams	推动多层网络训练成为可能
1998	LeNet	Yann LeCun 等	在手写数字识别中验证卷积网络有效性
2012	AlexNet	Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton	引爆深度学习视觉浪潮
2015	ResNet	Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (MSRA)	解决深层网络退化问题,成为视觉主干常用选择
2017	Transformer	Ashish Vaswani 等 (Google)	提出自注意力机制,变革序列建模
2020s	多模态模型	多机构并行探索 (OpenAI CLIP, GPT-4V 等)	推动视觉、语言、动作的统一建模
2025	3D 世界模型	李飞飞团队 (World Labs)	发布最新 3D 世界模型,推动具身智能与可预测世界表示

注：粗体为人工智能发展重要人物，将被重点讲解。

1.1.3 深度学习三件套：模型 / 损失函数 / 优化器

深度学习可以理解成“可训练的函数”。它由三件套构成：模型、损失函数、优化算法。这三者决定了模型“能表达什么”“学什么目标”“怎么学”。

- 模型：线性层提供可学习的特征组合，但只能表示线性关系；激活函数（ReLU/Sigmoid/Tanh）引入非线性，才能学到复杂模式。
- 损失函数：训练的目标不是“准确率”，而是最小化损失；分类任务常用交叉熵，回归任务常用 MSE。

- 优化器：SGD 最基础稳定但可能收敛慢；Adam 适合新手、默认好用。

1.2 WHY：为什么要学深度学习？

1.2.1 深度学习在智能驾驶中的核心地位

深度学习在智能驾驶中扮演核心角色，主要体现在以下几个方面：

- 感知能力：深度学习驱动的视觉和点云处理模型（如 CNN、PointNet）是实现环境感知的基础。
- 预测与规划：深度学习模型（如 RNN、Transformer）用于预测其他交通参与者的行为，支持安全规划。
- 端到端学习：深度学习使得从传感器输入到控制输出的端到端学习成为可能，简化了系统架构。
- 多模态融合：深度学习支持多传感器数据（摄像头、雷达、激光雷达）的融合，提高感知的鲁棒性和准确性。
- 持续改进：通过大规模数据训练和在线学习，深度学习模型能够不断提升性能，适应复杂多变的驾驶环境。

1.2.2 深度学习的应用场景

在智能驾驶中，深度学习的主要应用场景及对应模型包括：

- 目标检测与识别：使用 CNN 模型（如 YOLO、Faster R-CNN）检测车辆、行人、交通标志等。
- 语义分割：利用深度学习实现道路、车道线、障碍物的像素级分类。
- 行为预测：通过 RNN 或 Transformer 模型预测其他车辆和行人的未来轨迹。
- 路径规划：深度强化学习用于生成安全、高效的行驶路径。
- 驾驶决策：端到端模型直接从传感器数据学习驾驶策略。
- 多传感器融合：结合摄像头、雷达和激光雷达数据，提高环境感知的准确性和鲁棒性。

1.3 HOW：如何搭建最小训练闭环（来自 PyTorch Tutorial）

1.3.1 一个可复现的训练闭环

- 数据准备（Dataset/Dataloader）
- 模型定义（nn.Module）
- 损失函数与优化器
- 训练循环（forward -> loss -> backward -> step）
- 评估与保存

最小训练循环（示意）：

```
for images, labels in dataloader:
    outputs = model(images)
    loss = criterion(outputs, labels)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

理解这个闭环后，你就能在第 2 章跑通“CNN vs ResNet18”的实操对比。

1.3.2 过拟合与泛化基础

训练并不等于“学会”。真正重要的是模型能否在新数据上表现稳定。
因此你要理解“过拟合”和“泛化”的基本判断逻辑。

新手泛化判断规则：

- 训练 loss 下降、验证 loss 同步下降：正常学习
- 训练下降、验证不下降：可能过拟合
- 训练与验证都不下降：学习率不合适或模型太弱

避免过拟合的最小手段：

- Dropout：训练时随机“关闭”部分神经元
- Weight Decay：限制参数过大
- 数据增强：用更多“视角”提升泛化

第 2 章 实操深度学习案例（Day 1 下午）

本章基于 PyTorch Tutorial 的训练结构，完成“CNN vs ResNet18”对比实验。
目标是把第 1 章的理论变成可执行的训练闭环。

2.1 你将学到什么

对于一名深度学习初学者来说，本章实操能帮助你达成以下可验证的目标：

- 能用一句话描述本实验的目标与数据（例如：在 CIFAR-10 子集上比较两种模型的泛化表现）。

- 能运行训练脚本并成功生成训练日志、模型权重与训练/验证曲线图。
- 能读懂并解释训练/验证的损失与准确率曲线，识别过拟合或欠拟合现象，并提出至少一种改进方案。
- 学会并能实际应用至少三种常用的泛化增强手段（如数据增强、Weight Decay、Dropout 或早停）。
- 能比较 Simple CNN 与 ResNet18 的训练时间、参数规模与测试性能，并解释差异产生的主要原因。
- 能在一次小实验中修改一个超参数（例如学习率或是否使用数据增强），记录并解读结果差异。

（以下开发过程略过）

2.2 实验设计（来自 PyTorch Tutorial + CIFAR 实战范式）

2.2.1 数据集：CIFAR-10

2.2.2 对比模型

2.2.4 实验配置

2.3 实操步骤（增强版，可验证且可重复）

2.4 评估与可视化

2.5 误差分析（从指标到样本）

2.6 小型改进实验（只改一个变量）

第 3 章 探索智能驾驶模型（Day 2 上午）

3.1 你将学到什么

把智能驾驶前沿概念与本次实操闭环对齐，理解“真实系统中这些概念的位置”。

3.2 世界模型与智能驾驶（核心概念）

世界模型强调“从当前状态预测未来状态”。

在智能驾驶中，没有预测就没有安全规划，因此世界模型是关键能力之一。

3.2.1 最小世界模型定义

- 状态 s_t ：当前世界状态表示（可由感知模型输出）
- 动作 a_t ：车辆或系统动作
- 转移：预测 s_{t+1} （未来状态）

最小输入/输出定义：

输入是连续 T 帧的状态表示，输出是下一时刻的状态或场景变化。

3.2.2 智能驾驶中的直接映射

- 前车减速 -> 预测距离变化
- 行人横穿 -> 预测占用区域变化
- 弯道进弯 -> 预测车道曲率变化

本课程实操不强求复杂世界模型实现，但要求理解“为什么必须预测”。

3.2.3 世界模型发展脉络（概念级）

- RSSM / Dreamer / PlaNet：用潜变量建模时序状态
- 模型式强化学习：先学世界，再做决策
- 李飞飞的世界模型倡议：推动“可生成、可预测”的世界表示

3.2.4 与实操的对应

本课程第 2 章训练的 CNN/ResNet 可以视为“感知特征抽取器”。

在世界模型任务中，你会将每一帧的特征作为状态表示，再去预测下一时刻状态。

也就是说：第 2 章解决“看见什么”，为本章的“预测未来”提供输入基础。

检查点：

- 能用 2 句话解释世界状态与世界模型
- 能说清本课程的输入与输出

3.3 智能驾驶技术发展里程碑（历史视角）

早期阶段：规则与传感器（1980s-2000s）

深度学习进入视觉感知（2012-2016）

多传感器融合与 BEV 发展（2017-2020）

预测与世界模型兴起（2018-2022）

语义与多模态趋势（2021-2024）

工程化与闭环迭代（2016-至今）

3.4 概念对照

先给出一条主线：从数据与感知开始，经过融合与表示，建立世界模型预测，再进入决策与工程落地，最终服务于“可预测世界状态”的目标。

实施顺序为：数据/基准层 -> 感知/检测层 -> 融合层 -> 表示层 -> 预测层 -> 语义层 -> 决策层 -> 工程层 -> 目标层。

分层架构图（概念 vs 技术栈，一表读懂）

分类	概念层级（作用）	技术栈示例（非唯一）
基础与保障	数据/基准层：训练与验证数据	BehavioralCloningTrackData / CIFAR-10
感知与表示	感知/检测层：看见什么	YOLO / Faster R-CNN / BEVDet / ResNet18
感知与表示	融合层：如何对齐信息	多视角融合 / 时序对齐 / 卡尔曼滤波
感知与表示	表示层：怎么表示世界	BEV Encoder / Occupancy / Feature Grid
预测与决策	预测层：如何看未来	RSSM / Dreamer / PlaNet / Simple CNN
感知与表示	语义层：如何解释与约束	CLIP / VLM / VLA
预测与决策	决策层：如何使用结果	规划器 / 策略网络 / MPC
基础与保障	工程层：训练->推理->导出	PyTorch / torchvision / TensorFlow / Keras / ONNX / TensorRT
基础与保障	目标层：统一可预测世界状态	表征学习 / 系统建模框架

注：教案提供实操案例的技术栈示例以加粗表示。

3.4.1 感知融合

3.4.2 BEV/3D

3.4.3 世界模型时序建模

3.4.4 VLM/VLA

3.4.5 CLIP

3.4.6 状态驱动控制

3.4.7 工程化流程

3.5 技术路线对比

以下对比用于帮助建立“路线分类”概念，不涉及商业评价或性能结论。你可以将其理解为不同系统组织方式的选择。

- 传统模块化方案（感知/预测/规划分工）：可解释性强、工程边界清晰，便于独立验证与替换。
- 多传感器融合路线（相机+激光雷达+雷达）：强调冗余与稳定性，工程复杂度与成本更高。
- 纯视觉路线：强调软件闭环与规模化，依赖更强的数据与训练策略。
- 端到端倾向路线：强调从感知到控制的统一学习，依赖大规模数据闭环与持续迭代。

典型路线示例（非结论性）：

- Waymo - Waymo Driver：多传感器融合路线代表。
- 小鹏 - XNGP：融合路线与工程化落地并行推进。
- 华为 - ADS：融合路线与工程化落地并行推进。
- 理想 - AD：融合路线与工程化落地并行推进。
- 特斯拉 - FSD：纯视觉与端到端倾向路线代表。

第 4 章 实操智能驾驶案例（Day 2 下午）

4.1 你将学到什么

本章完成“行为克隆”从数据准备、训练到评估的完整闭环，对应智能驾驶的最小端到端落地实操。

4.2 数据与环境准备

4.2.1 数据集介绍：Udacity Behavioral Cloning Track Data

为什么选择该数据集作为实操案例：

- 真实驾驶数据：数据来源于 Udacity 自动驾驶模拟器，包含真实驾驶场景
- 任务明确：行为克隆任务清晰，适合入门级深度学习实操
- 数据量适中：完整数据集约 8000 条记录，适合快速实验
- 多模态数据：包含图像与车辆状态，便于理解多模态输入
- 开源可用：数据集公开，易于获取与使用

(以下开发过程略过)

4.3 训练步骤（行为克隆）

4.4 评估与可视化

4.5 评估指标解读

4.5.1 MSE 的含义

4.6 基于 CLIP 的语言条件相似度评分示例

课程总结与交流

- 复盘两天所学内容，梳理深度学习与智能驾驶的核心知识体系
- 分享学习资源与后续提升路径
- 答疑与交流，帮助学员解决实际问题