

深度学习纳米学位课程大纲

学习 Pytorch、Keras、TensorFlow 框架等主流技术，成为深度学习工程师



更新日期 / 2019-02-19

学习目标

当你完成深度学习纳米学位所有项目之后，你将拥有娴熟的 Python 编程能力，并熟练掌握神经网络知识脉络。你将为继续深造自然语言处理工程师、计算机视觉工程师、无人驾驶纳米学位、深度强化学习、机器人开发等高阶项目打下坚实基础，成为大数据、人工智能领域的稀缺人才。

整个深度学习纳米学位**一共由 5 个实战项目和超过 30 个在线 Lab 组成**。在每个项目中，你都会通过实战演练，获得领域专家的逐行代码审阅和反馈，学习硅谷先进的技术标准，与硅谷编程开发者的思维同步学习。每个项目的复杂程度不一，每个项目都是在一个模块课程的学习结束之际才开始的，让你将吸收的知识活学活用。

先修知识

你需要具备编程知识、统计学知识、微积分和线性代数知识，熟悉 Python 的常用类库（Numpy, Matplotlib 等）。如果你没有以上这些基础，建议你先学习我们的 [Python 人工智能入门](#) 纳米学位。

项目时长

这个课程长达 21 周，平均需要每周花 10 小时左右的时间。

学习服务

详情请咨询学习规划师。

第 1 部分：神经网络

课程内容

深度学习简介

学习目标	学习技能
欢迎学习此课程	深度学习技术的广泛应用前景 认识导师 你将会完成的项目 准备工作
应用深度学习	深度学习简介 风格迁移 DeepTraffic, Flappy Bird 中的深度学习介绍 深度学习课外图书推荐
Anaconda	Anaconda 是什么？ 如何安装 Anaconda 如何管理包 如何管理环境 更多环境操作 Anaconda使用的最佳做法 Python 版本说明
Jupyter Notebooks	Jupyter notebook 是什么？ 安装 Jupyter notebook 启动 notebook 服务器 notebook 界面 代码单元格 Markdown 单元格 如何使用快捷键 Notebook 中的 Magic 关键字 转换 notebook 成为其他格式 如何创建幻灯片
矩阵数学和 NumPy 复习	简介 理解数据维度 NumPy 中的数据 元素级矩阵运算 NumPy 中的元素级运算 NumPy 中的矩阵乘法 NumPy 中的转置 NumPy 练习

神经网络

学习目标	学习技能
神经网络简介	神经网络分类问题 神经网络线性界线 感知器及何为“神经网络”？ 在线Lab：感知器算法实现及解答 非线性界线及误差函数 误差函数与梯度下降 离散型与连续型 - 为什么使用sigmoid函数 多类别分类与Softmax 函数 One-Hot 编码 最大似然率 最大化概率 交叉熵 - 损失函数 多类别交叉熵 Logistic 回归 在线Lab：梯度下降及实现 非线性数据、模型及神经网络结构 神经网络的前向及反向传播 在线Lab：用神经网络进行学生人群分类
线性回归	在线Lab：房价预测 绝对值技巧 & 平方技巧 梯度下降及误差计算（平均绝对值误差、平均平方误差） 在线Lab：最小化误差函数练习 - 预测生命长度 绝对值误差 VS 平方误差 高维度解数学方程表组 多项式回归及正则化 神经网络中的回归
实现梯度下降	平方平均误差函数 在线Lab：梯度下降实现梯度 多层感知器及反向传播实现 神经网络进阶阅读
训练神经网络	训练优化及测试 过拟合和欠拟合 早期停止 正则化及Dropout 局部最低点 随机重新开始 何为梯度消失 其他激活函数 Batch 和随机梯度下降 学习速率衰退

	动量及其他误差函数
在线实验室:情感分析	通过社交媒体中的评分数据及评论，对评论文本进行文本操作及分析。该实验将完成文本模型处理的全流程，包括模型的问题确认、数据预处理、模型搭建、模型优化及训练。经过该实验室，你将进一步了解数据文本处理、神经网络训练及工业问题的实际解决。
TensorFlow	介绍 安装 TensorFlow 在线 Lab : TensorFlow 输入、TensorFlow 数学 在线 Lab : TensorFlow 里的线性函数、TensorFlow Softmax 在线 Lab : TensorFlow 交叉熵、Mini-batch 及 Epochs 在线 Lab : TensorFlow 实现NotMNIST，包括：两层神经网络、TensorFlow ReLUs、保存和读取 TensorFlow 模型、参数微调、TensorFlow Dropout及总结
Keras	Keras介绍 在线 Lab : Keras 神经网络实现录取学生预测及优化程序 在线 Lab : 电影数据预测-IMDB
深度学习工具Pytorch	Pytorch实现单层神经网络 使用矩阵乘法构建网络及多层网络实现 在 PyTorch 中构建神经网络及实现 Pytorch 实现 Softmax PyTorch 中的网络架构实现 Pytorch 训练网络 在线 Lab : 分类 Fashion-MNIST，包括：推理和验证，dropout、保存和加载模型、加载图像数据、启用 GPU、迁移学习及其他注意事项

实战项目: 预测共享单车使用情况

从零开始搭建并训练一个神经网络，并用该网络预测每日自行车租客人数量，为某一共享单车预测某一天内需要的使用量，帮助他们作出管理自行车的决策。数据库中包包含 2011 年 1 月 1 日到 2012 年 12 月 31 日期间每天每小时的骑车人数，同时也包含季节、天气、月份等影响骑行人数的数据。你将自己构建一个后向传递的神经网络，并在此基础上调整学习速率、迭代次数、隐藏节点等参数，训练自己的神经网络，用以预测“未来”某一天的骑行人数，并与该天的实际骑行人数进行比较。通过这个时间序列数据集，你将实现并掌握你自己的后向传播神经网络。

第 2 部分：卷积神经网络

课程内容

学习目标	学习技能
卷积神经网络及Pytorch实现	CNN（卷积神经网络）的应用 计算机如何解析图像 损失和优化 在 PyTorch 中定义网络、训练网络

	在线 Lab : MLP （多层神经网络）分类、MLP 分类 MNIST 实现、模型验证、损失计算 卷积神经网络的局部连接性、过滤器和卷积层、步长和填充, 池化层 在线 Lab : 寻找边缘 在线 Lab : 层级可视化 PyTorch 中的卷积层, 特征向量实现 在线 Lab : CNN 用于 CIFAR 图像分类 , 包括分类示例、PyTorch 中的 CNN、图像增强、突破性的 CNN 结构、可视化 CNN 及总结
Keras实现卷积神经网络	Keras 中的模型验证 在线 Lab : MNIST 手写数字分类 , 包括 : 训练 MLP、Keras 中的卷积层、Keras 中的最大池化层、Keras 中的 CNN、Keras 中的图片增强功能、用 Keras 进行迁移学习
在Tensorflow中的卷积神经网络	在线 Lab : TensorFlow 卷积层 在线 Lab : TensorFlow 最大池化及各池化机制 在线 Lab : TensorFlow 中的1x1 卷积及 Inception 模块
权重初始化	权重初始化简介 1 和 0 、均匀分布、正态分布的差别 补充阅读
自编码器	自编码器课程简介 自编码器实现 卷积自编码器实现
迁移学习	迁移学习简介 在线 Lab : 用 VGGNet 进行迁移学习及实现 在线 Lab : 数据预处理实现 在线 Lab : 分类器实现 在线 Lab : 训练模型实现
癌症检测深度学习 (Sebastian Thrun)	介绍皮肤癌及皮肤癌生存概率 医学分类介绍 在线 Lab : 实现皮肤癌分类 , 包括训练神经网络、随机权重与预初始化权重、训练验证、敏感性与特异性、癌症诊断、ROC 曲线, 将我们的结果与医生诊断结果相比较及可视化

实战项目: 狗狗品种识别

设计并训练一个卷积神经网络 (CNN), 来分析狗狗图像, 并据此准确区分它们的品种。当你将一只狗狗的图像输入你的算法, 它将被识别并估计为狗的品种, 如果输入的图像是人, 你的模型将识别为最相近的狗的品种。使用迁移学习和其他知名架构来优化这一模型——为更进阶的应用做好准备! 当你完成这一项目时, 你需要将一系列模型拼接在一起, 了解每个模型的优点与缺点。你会理解, 你设计的最终目标是为用户带来愉快的体验!

第 3 部分：循环神经网络

课程内容

学习目标	学习技能
循环神经网络	循环神经网络简介、历史及应用 前馈神经网络及反向传播算法 循环神经网络前馈实现 基于时间的反向传播算法及练习 循环神经网络总结及小结
长短期记忆网络 (LSTM)	LSTM 介绍 RNN vs LSTM LSTM 基础及架构 学习门, 遗忘门, 记忆门, 应用门 其他架构及 LSTM 总结
实验：实现RNN 和 LSTM	在线 Lab：实现 RNN, 时间序列预测 , 包括：训练和记忆功能、字符 RNN、序列分批、批处理数据、定义模型并做出预测
超参数	学习率 Minibatch 大小 训练迭代 / Epoch 的次数 隐藏单元/层的数量 RNN 超参数
嵌入和 Word2vec	嵌入实现 Skip-gram 实现
情绪预测 RNN	在线 Lab：情绪 RNN , 包括：数据预处理, 创建测试集, 搭建 RNN, 训练 RNN
注意力模型	注意力简介 编码器和解码器, 序列到序列 注意力编码器、注意力解码器 Bahdanau 注意力与 Luong 注意力, 乘法注意力, 加法注意力, 加法和乘法注意力 注意力模型在计算机视觉应用及其他注意力方法 在线 Lab：注意力模型实现
股票价格预测	在线 Lab：时间序列模型预测股票价格 通过股票价格预测案例, 进一步延展学习“股票价格投资”的背景、理论及数据特征。本案例将通过时间序列模型, 来了解经典统计学及概率机器学习模型之间的差异。基于这一股票价格预测的案例, 你将比较不同时间序列模型在股票价格预测上的差异及优劣。

实战项目：生成电视剧剧本

在 TensorFlow 上, 使用 RNN 创作你自己的《辛普森一家》电视剧剧本。你将会用到《辛普森一家》第 27 季中部分剧本的数据集。你创建的神经网络将为一个在 Moe 酒馆中的场景生成一集新的剧本。在该项目中, 你将学习分

词、GPU调用、检查点创建、不同 batch 学习等内容。该RNN模型将成为你学习“自然语言处理”的第一步，不知道你的 RNN 神经网络会学习到一个怎样的故事呢？

第 4 部分：生成对抗网络

课程内容

学习目标	学习技能
生成对抗网络	介绍 Ian Goodfellow 你可以用 GAN 做些什么？ GAN 的运作原理, 博弈论和均衡 训练 GAN 的实用技巧 在线 Lab：构建 GAN , 包括：生成器网络, 辨别器网络, 训练损失, 训练优化器, GAN 模型优化
深度卷积生成对抗网络	深度卷积 GAN DCGAN 架构 批次归一化 在线 Lab：DCGAN 实现 , 包括：DCGAN 生成器和辨别器, 构建和训练网络, 超参数设置
半监督学习	半监督学习 半监督 GAN 分类器 在线 Lab：使用街景数据库生成门牌号 ；准备数据, 建立生成器和辨别器, 模型损失练习, 模型优化练习, 训练网络, 半监督 GAN 训练结果

实战项目: 人脸生成

打造一对多层神经网络，通过使他们相互对抗来生成真实人脸。在项目中你将获得一系列明星人脸，试试看用他们能生成什么样的新面孔吧！你将通过让你的神经网络学习 MNIST、CelebA 数据集，学习图像数据处理的方式。并通过辨别器、生成器、损失函数、优化函数的构建，帮助你的对抗网络互相学习，最终达到能够自动生成你需要的图像！快来看看你的网络生成的明星脸吧！

第 5 部分：深度强化学习

课程内容

学习目标	学习技能
欢迎学习强化学习	简介及应用 OpenAI Gym资源

	参考指南
强化学习框架：问题	简介 设置 - 回顾 阶段性任务与连续性任务 奖励假设, 目标和奖励, 累积奖励, 折扣回报 MDP动态平衡实现
强化学习框架：解决方案	简介及策略 状态值函数及贝尔曼方程 最优性及动作值函数 最优策略
动态规划	简介 迭代方法 迭代策略评估 动态规划, 包括动作值实现, 策略改进实现, 策略迭代实现, 截断策略迭代实现, 值迭代实现, 知识检验及总结
MC 蒙特卡洛方法	简介 MC 包括状态值实现, 动作值实现, 广义策略迭代, MC 增量均值, MC 策略评估, MC 策略改进, Epsilon 贪婪策略, 探索与利用实现, 常量 α 实现, 蒙特卡洛方法总结
TD 时间差分方法	简介 TD(0)实现, 动作值, Sarsa(0)实现, Sarsamax实现, 预期 Sarsa实现, 分析性能及总结
迷你项目：解决 OpenAI Gym 的 Taxi-v2 任务	在线 Lab：出租车线路优化
连续空间中的强化学习	深度强化学习 离散空间与连续空间 空间表示法 - 离散化, Tile Coding, Coarse Coding 函数逼近 - 线性函数逼近, 内核函数, 非线性函数逼近
深度Q-学习	深度 Q 学习简介 神经网络作为值函数, 蒙特卡洛学习, 时间差分学习 Q 学习 深度 Q 网络, 包括：经验回放, 固定 Q 目标, 深度 Q 学习算法, DQN 改进, 实现深度 Q 学习, TensorFlow 实现
策略梯度	基于策略的方法简介 为何要使用基于策略的方法？ 策略函数逼近 随机性策略搜索 策略梯度, 蒙特卡洛策略梯度, 受限策略梯度
行动者-评论者方法	行动者-评论者方法 两个函数逼近器, 行动者和评论者 优势函数, 行动者-评论者方法与优势函数
在线实验室：机器学习模型部署	机器学习/深度学习模型部署使你能够使用训练有素的模型来分析新的用户输入。构建模型, 部署模型, 并创建从网站访问模型的网关。该在线实验室将带你完善模型预处理、训练、调参、部署（云部署）的全流程, 通过一个波士顿房价预测

实战项目：训练四轴飞行器学会飞行

设计一个深度强化学习系统，来控制你的四轴飞行器的学会飞行。你将通过构建你自己的惩罚函数、强化学习模型、深度学习隐藏层来帮助你的四轴飞行器了解他的每一个动作的优劣。你的四轴飞行器将从一系列动作状态中，选择最优的策略来平稳起飞和降落。在该模型中，你需要调整超参数模型来获得较好的学习效果。通过该项目，你会进一步加深对深度神经网络的理解。