前言

Its goal is to give you the concepts, the intuition, and the tools you need to actually implement programs capable of learning from data.

本书目标是教会你从数据中学习的概念，直观感受，和一些你需要用到的工具。

本书代码库

<https://github.com/ageron/handson-ml>

本书要求：

懂得基本python语法

熟悉Numpy，Pandas，Matplotlib

数学要求：

calculus, linear algebra, probabilities, and statistics

微积分，线性代数，概率论，统计学

RoadMap

本书分为两部分：

**第一部分**

* Ml是什么？解决什么问题？ML系统的主要领域和关键概念是什么？
* 一个典型MP项目的主要步骤
* 通过调试模型来学习
* 优化损失函数
* 处理，清洗和准备数据
* 选择和设计（创建）特征
* 选择一个模型并通过交叉验证法调参
* ML的核心挑战，尤其是过拟合和欠拟合（偏差和方差折衷）
* 减少训练数据的维度，以避免维度灾难
* 最常见的算法:

Linear and Polynomial Regression, 线性和多项式回归

Logistic Regression,

k-Nearest Neighbors,

Support Vector Machines,

Decision Trees, Random Forests，

Resemble methods.

**第二部分**

神经网络和深度学习，包含以下主题

* 什么是神经网络？有什么好处？
* 使用tensorflow来建立和调试神经网络
* 最重要的神经网络架构：

Feedforward neural nets, 正反馈神经网络

Convolutional nets, 卷积神经网络

Recurrent nets, 周期网络

Long short-term memory (LSTM) nets 长短时记忆网络

Autoencoders 自编码器

其他学习资源推荐

Andrew Ng’s ML course on Coursera 吴恩达的课

Geoffrey Hinton’s course on neural networks and Deep Learning

ML博客文章：

<https://www.quora.com/What-are-the-best-regularly-updated-machine-learning-blogs-or-resources-available>

深度学习网站：

<http://deeplearning.net/>

<https://www.kaggle.com/>

**第一部分 The Fundamentals of Machine Learning**

**第一章.The Machine Learning Landscape 机器学习大观**

ML的定义：

ML is the science (and art) of programming computers so they can learn from data.

让机器能够从数据中学习。

机器学习类型

* 是否需要人工监督：

supervised（监督学习）,

unsupervised(非监督学习),

semisupervised（半监督）,

Reinforcement Learning(加强学习)）

* 是否能够联机递增（在线学习，批次学习）
* 是将新数据和已知数据作比较，还是根据训练数据建立模型（基于实例，还是基于模型）

**Supervised Learning 监督学习算法**

核心特征，有label。

分类问题经常用到。

常见的监督学习算法：

* k-Nearest Neighbors k-邻近算法
* Linear Regression 线性回归
* Logistic Regression 逻辑回归
* Support Vector Machines(SVMs) 支持向量机
* Decision Trees and Random Forests 决策树和随机森林
* Neural networks 神经网络

监督学习例子：

垃圾邮件拦截

**Unsupervised无监督学习算法**

**Clustering 聚类问题**

k-Means

Hierarchical Cluster Analysis (HCA)  层序聚类分析

Expectation Maximization 期望最大化

**Visualization and dimensionality reduction** **可视化和降维**

Principal Component Analysis (PCA) 主成分分析

Kernel PCA

Locally-Linear Embedding （LLE） 局部线性嵌入

t-distributed Stochastic Neighbor Embedding(t-SNE) t分布随机邻居嵌入

**Association rule learning** **关联规则学习**

**Apriori** 先验学习

无监督学习例子：

购物商场可视化处理数据

博客读者分析

常用见的3个无监督学习任务

Dimensionality reduction 降维也叫feature extraction：将相互有关系的几个特征合并到一个维度。

Anomaly detection 异常检测，比如检测到异常数据自动从数据集中删除。

Association rule learning 关联规则学习，主要目标是通过大量数据，发现其中的相关性。

**Semisupervised learning 半监督学习**

例如，图片检索识别，一大堆图片，通过非监督学习将图片根据不同的人脸分类，通过监督学习将这些已经分类好的图片，每一类做一个标识。

大多数半监督学习都是监督学习和非监督学习的结合。例如deep belief networks(DBNs)是基于restricted Boltzmann machines(RBMs)叠加。RNMs是在非监督下训练的，但是整个系统是通过监督学习技术调试的。

**Reinforcement Learning** **强化学习**

强化学习是一种非常不同的动物。学习系统在此上下文中称为agent，它可以观察环境，选择并执行操作，并获得相应的奖励(或者以负面奖励的形式进行惩罚，如图1-12所示)。然后，它必须自己学习什么是最好的策略，即所谓的政策，以便在一段时间内获得最大的回报。策略定义了代理在给定情况下应该选择什么操作。

例如，很多机器人通过强化学习学习走路。Alphago也是强化学习的一个例子。

**Batch learning VS Online learning**

**Batch learning 分批学习也叫offline learning**

因为每次训练的数据量庞大，通常非常耗时，几天甚至几周，因此一般是线下训练，训练后发布，如果有新类型的数据要训练就要重新训练和发布。对硬件设备要求也很高，因此如果是要求快速适应新的数据或者是没有很高的资本投入，这不是一种很好的选择。

**Online Learning 线上学习**

通过循序地添加数据实例训练系统，一个一个或者一小组一小组地添加，叫做最小批次。每一次学习都快速便宜，所以可以实现线上学习。

线上学习通过持续地接收数据，能够快速自动地改变。而且非常节约运算资源。

在线学习算法还可以用于，在无法装入一台机器主存的大型数据集上，对系统进行培训(这称为核心外学习)。该算法加载部分数据，对该数据运行一个训练步骤，并重复该过程，直到在所有数据上运行为止。这个过程都是在线下进行的，所以更规范的名字应该是渐进式学习。

学习率：用来规定系统学习新数据速度的一个参数。如果这个参数设置很高，那么系统学习新数据会很快，但是同时，也会很快忘掉老数据。相反，如果很低，那么会学的很慢，同时对新数据里的噪音不敏感。

对于在线学习的一个巨大挑战是坏数据会被系统学习，从而降低系统的表现。如果是线上，用户会能够感受的到。为了减少这种风险，你需要能够回退到之前的状态，并且需要管理输入的数据，增加异常检测功能（通过异常检测算法）。

**Instance-based VS Model-Based Learning基于实例和基于模型的学习**

另一种分类ML学习的方式，是通过区分系统如何泛化。大多数机器学习的任务是关于预测。这就意味着系统要通过给出的数据例子，来预测没有见过的情况。在训练集上表现良好很好，但是这不重要，真实的目标是能够在新实例上表现良好。

主要有两种泛化途径：基于实例和基于模型。

**Instance Based**基于实例

系统记住这些例子，然后使用相似度度量将其推广到新的案例。

两封电子邮件之间的一个(非常基本的)相似度度量方法是计算它们共有的单词数量。如果电子邮件与已知的垃圾邮件有许多相同的单词，系统会将其标记为垃圾邮件。

**Model Based基于模型**

从一组例子中归纳的另一种方法是为这些例子建立一个模型，然后使用它

模型进行预测。

衡量模型的性能，可以通过定义效用函数(utility function)（适应性函数 fitness function，衡量模型有多好。），或者可以通过定义损失函数(cost function),衡量系统有多糟糕。大多数通过损失函数，衡量预测结果与实际结果的差距；主要目标是减少差距。

41页，一个手写例子。

**机器学习的主要挑战**

**训练数据量不足**

**不具代表性的数据**

**训练数据质量太差**

* **当有些实例明显是异常数据时，直接去掉这些数据**
* **如果有些实例缺少一些特征时，忽略这些特征或者时填补这些缺少的特征**

**互不关联的特征**

* **特征选区：从现有的特征中选取最有用的特征**
* **特征提取，降维算法**
* **通过新的数据，创造新的特征**

**过拟合**

出现过拟合可能的解决办法（regularization）

* 通过选择参数较少的模型(例如，线性模型而不是高次多项式模型)，通过减少训练数据中的属性数量或约束模型来简化模型
* 收集更多的训练数据
* 减少训练数据集中的噪音数据

在学习过程中正则化的值可以由（hyperparameter）超参数控制。超参数是学习算法的参数(不是模型的参数)。因此，它不受学习算法本身的影响;它必须在训练之前设置，并且在训练期间保持不变。如果将正则化超参数设置为一个非常大的值，就会得到一个几乎平坦的模型(斜率接近于零);学习算法几乎可以肯定不会过度拟合训练数据，但它不太可能找到一个好的解决方案。超参数整定是机器学习系统的重要组成部分。

**欠拟合**

模型函数太简单，导致模型不能满足数据的结构。

解决欠拟合的主要方法：

* 选择一个更强大的模型，更多的参数
* 填入更好的特征给算法（feature engineering）
* 减少模型的约束（减少正则化超参数）

**回顾总结**

* 机器学习通过从数据学习，而不是硬编码
* 有很多机器学习系统：监督非监督，批处理和在线，基于实例和基于模型，等等
* 通过把训练数据集喂给算法来让算法学习。如果是基于模型的需要调整参数来让模型表现良好。如果是基于实例的，系统记住这些例子，然后使用相似度度量将其推广到新的案例。
* 如果训练数据集太小或者数据质量太差。系统都会表现很差。（欠拟合和过拟合）

**测试和验证**

训练集training set 80%和测试集testing set 20%。

Generalization error 泛化错误

如果模型的训练错误很低但是泛化错误很高，表示模型在训练集上过拟合了。