

数据竞赛
数据科学

Kaggle竞赛 入门讲义

竞赛知识都在这里~

by
阿水

目录

1	引言	7
1.1	课程目标	7
1.2	课程受众	8
1.3	课程 SMART 原则	8
1.4	课程 SQ3R 学习法	8
1.5	课程基础	9
2	数据科学必知必会	10
2.1	什么是数据科学?	10
2.2	为什么学习数据科学?	11
2.3	如何学习数据科学?	11
2.4	数据科学包含的知识点	12
2.5	数据竞赛平台	14
2.6	竞赛实例讲解	14
2.6.1	Rong360-用户贷款风险预测	16
2.6.2	Planet: Understanding the Amazon from Space	16
2.7	本章小结	16
3	机器学习基础	18
3.1	误差与过拟合	18
3.2	模型评价方法	19
3.3	偏差与方差	21
3.4	线性模型	22
3.5	树模型	22
3.6	KNN 模型	23
3.7	神经网络	24
3.8	本章小节	24
4	常见的树模型	25
4.1	Isolation Forest	25
4.2	随机森林	25
4.3	GBDT	25
4.4	XGBoost	25
4.5	LightGBM	25
4.6	CatBooost	25
4.7	本章小结	26
5	深度学习	27
5.1	基本原理	27
5.1.1	全连接网络	28
5.1.2	深度学习的正则化	29

5.1.3	深度学习的优化	30
5.2	卷积神经网络	30
5.3	循环神经网络	31
5.4	自编码器	32
5.5	深度学习的实践论	32
5.6	本章小节	33
6	数据挖掘的工作流程	35
6.1	数据分析	35
6.1.1	赛题背景分析	36
6.1.2	赛题数据分析	37
6.2	特征工程	38
6.2.1	数据清洗	38
6.2.2	特征预处理	39
6.2.3	特征提取	40
6.2.4	特征筛选	40
6.3	训练与验证	41
6.4	模型融合	44
6.5	本章小结	45
7	结构化数据挖掘	46
7.1	常规类型比赛	46
7.1.1	Two Sigma Connect: Rental Listing Inquiries	46
7.2	CTR 类型比赛	46
7.2.1	WSDM-百度好看 APP	46
7.2.2	Outbrain Click Prediction	47
7.3	用户信息预测	47
7.3.1	TalkingData Mobile User Demographics	47
7.3.2	易观-用户性别年龄预测	47
7.4	匿名数据挖掘比赛	47
7.4.1	中诚信征信比赛	47
7.4.2	Allstate Claims Severity	48
7.4.3	Porto Seguro's Safe Driver Prediction	48
7.5	本章小结	49
8	计算机视觉任务	50
8.1	数字图像处理	50
8.2	图像特征算子	51
8.3	常见的视觉任务	52
8.3.1	图像分类	52
8.3.2	图像检索	53
8.3.3	物体检测	55

8.3.4	字符识别	55
8.4	视觉类型比赛	55
8.4.1	Quick, Draw! Doodle Recognition Challenge	56
8.4.2	Google Landmark Retrieval Challenge	56
8.4.3	Google Landmark Recognition Challenge	56
8.4.4	TinyMind 人民币面值 & 冠字号编码识别挑战赛	57
8.4.5	Urban Region Function Classification	58
8.5	本章小节	58
9	自然语言处理任务	60
9.1	文本分词与词性标注	62
9.2	文本预训练技术	62
9.3	常见的 NLP 任务	63
9.3.1	文本分类任务	63
9.3.2	文本信息抽取	63
9.4	NLP 类型比赛	63
9.4.1	第三届阿里云安全算法挑战赛	63
9.4.2	第二届易观算法大赛-性别年龄预测	63
9.4.3	科大讯飞-大数据应用分类标注挑战赛	63
9.5	本章小节	63
10	其他相关任务	65
10.1	AutoML	65
11	侃侃而谈的博客	68
11.1	深度自编码器	68
11.2	模型的可解释性	68
11.3	深度学习框架的选择	68
11.4	深度学习的发展	70
11.5	人工智能的边界是什么?	71
11.5.1	如何选择一个合适的数据竞赛?	72
12	总结与展望	73
12.1	知识点总结	73
12.2	就业建议	73
13	附录	74
13.1	推荐书单	74
13.2	推荐链接	75
13.3	推荐公开课	75
13.4	常用 Python 库	76
13.5	比赛经验	77
13.6	常见面试题	78

13.6.1 常规面试题	78
13.6.2 机器学习面试题	79
13.6.3 计算机视觉面试题	80
13.6.4 自然语言处理面试题	80
13.7 常见数据岗位	80
13.7.1 数据分析师	80
13.7.2 数据挖掘 (DM) 工程师	81
13.7.3 机器学习 (ML) 工程师	81
13.7.4 图像 (CV) 算法工程师	82
13.7.5 文本 (NLP) 算法工程师	82
13.8 名词中英对照表	83

序

在进行撰写专栏文章和开设数据挖掘和数据竞赛的课程期间,我经常反复整理课程资料,导致同一个知识点可能多次整理。这样不仅浪费了我的个人时间,而且整理的知识点也没有体系非常零散,因此在 2019 年春节期间我尝试将数据科学相关的知识点进行了整理,并整理得到《数据科学公开课》讲义。

数据科学是一个快速发展的领域,几乎每周都会有新的模型和新的应用。数据科学也改变了就业环境,现如今企业更需要的是新时代下的精通数据科学的算法人才。数据科学包含的内容众多,且具有较强的实践色彩。**理论联系实际,边做边学是学习数据科学最有效的方式。**其中数据科学竞赛是最带有魅力色彩的一项:数据竞赛减少了领域壁垒,各行各业的从业者都可以参与进来;数据竞赛加快了领域发展速度,增强了从业者的交流。在工作过程中我发现,工作任务和数据竞赛也是类似的。在工作中有优化目标,有相应的规则和评分方式,这些都是和竞赛一一对应的,所以擅长数据竞赛的同学一般动手能力都很强。

数据科学包含的内容非常多而且杂,很难将知识点弄一个列表让大家依次学习。但终归来说,我们对计算机程序的期望都是一致的:**在给定输入的情况下,希望程序能够快速并正确的给出输出结果。**终归到底还是要写代码,写好代码并把代码写正确,最终解决问题。如何把代码写好,写快,写正确,是程序员始终的目标。

本讲义关注于数据科学的知识点,特别是数据竞赛的案例和解决方案,讲义的内容侧重解决流程和解决思路。我希望当你学习完讲义的内容后能够对数据科学以及数据挖掘的流程有一个清晰的认识,对数据应用场景和常见的解决方法有一定的了解,并最终能够动手解决对应的问题。举例来说当你遇到一个从图片识别字符的任务,你应该将其抽象成一个 OCR 问题,并优先使用成熟的 OCR 模型和数据集来完成解决。

讲义中的所有比赛的代码和资料我们都做了保存¹。我们精心整理了各种资料,欢迎扫描关注我们的公众号。最后希望你能够学有所成,希望讲义等资料能够帮助到你。



¹<https://github.com/datawhalechina/competition-baseline>

1 引言

同学们大家好，这里是《数据竞赛必知必会》，我是阿水。近年来数据挖掘/机器学习/深度学习的越加深入人心，“大众创业，万众创新”项目推动了人工智能的创业热潮，国内互联网公司也纷纷设立了人工智能部门，教育部甚至将在中小学开设了人工智能的课程。但对于一个相关从业者（一个学生和一个数据挖掘的矿工）来说，我们就更加需要广泛接受新知识和新方法。

近年人工智能在学术界和工业界的都非常迅速，2018年图灵奖颁发给了深度学习的三驾马车（Geoffrey Hinton、YannLeCun 和 Yoshua Bengio），但深度学习本质的相关概念和流程早在上个世纪都已经完成完备。同时在国内外都有很多非常优质的公开课，比如斯坦福的 CS231N 和 CS224。这些公开课每年都会开设，都是非常优质的学习资源。但数据科学领域发展非常迅速，每年都会有模型、数据集和新方法的出现，所以这是非常适合学习和从事数据科学的时代。但数据科学本身就是一个新领域，是多个领域的交叉方向，且是以理论与实践紧密结合的学科。所以在学习数据科学的时候，学习者往往会遇到很多细节：比如软件的安装、机器学习代码如何实现、数据如何分析以及数据如何采样。这些问题曾经困惑我许久，因此我想把我自己的一些学习经验和一些从业经验传授给大家。

课程的具体内容是以数据科学为主，具体以数据挖掘流程、常用的机器学习算法以及实战任务三个部分组成。在实战任务中按照数据的类型，我又将任务分为：结构化数据、视觉数据和文本数据，分别对应了相应的章节。课程的实际是理论与具体场景相结合，在讲解知识点的同时穿插具体的应用场景。希望大家在学习本次课程后，能够有所收获。

1.1 课程目标

我们本次课程的目标是，希望大家在学习完课程后：

1. 入门数据挖掘并能够结合实际场景进行实践；
2. 掌握常见的机器学习算法；
3. 掌握深度学习的基础和应用场景；
4. 学习计算机视觉常见的任务；
5. 学习自然语言处理的任务；

最终能够独立参加一场数据竞赛，或能够独立完成一项数据挖掘任务。这些也是我对大家的一点期许：学习知识点不仅要弄清楚知识点的脉络，同时也需要弄懂知识的具体应用场景。虽然结构化数据、图像数据和文本数据虽然在存储形式上有所区别，但其可用的方法往往都是相同的。比如聚类算法在计算机视觉任务和无监督任务中都有出现，Word2Vec 方法在自然语言和结构化数据中都能发挥作用。

结构化数据、图像数据和文本数据是最常见的数据类型，对应了结构化数据挖掘、计算机视觉和自然语言处理三个方向。建议大家在学习的时候不要完全将其割裂开来，不同任务类型可以相互借鉴，也可以用相同的算法进行解决。

1.2 课程受众

课程目标的受众是：想要入门或者深入学习数据挖掘的学生/从业者。

在学习课程之前，希望你有 Python 语言、计算机编程的基础知识。Python 语言现在应用非常广泛，希望大家能好好掌握。在课程中我也会讲解一些常见的概念和基础知识，同时我也会在附录章节留一些有用的学习资源。

我们希望在顺利完成课程之后：您遇到数据任务能够自己动手分析数据；可以自己构建机器学习模型；可以分析模型的误差并进行改进。

1.3 课程 SMART 原则

我相信大家对课程都有不同的期待，希望自己能够在课程中学习到一些知识，掌握一些技能。为了完成这些目标，我会在讲义的每章开头设置总结了章节内容，同时在章节末尾添加了一些练习题。为了帮助大家能够专注自己的目标，我推荐大家使用 SMART 原则来实现自己的目标。

SMART 原则 (S=Specific、M=Measurable、A=Attainable、R=Relevant、T=Time-bound)：

- 目标是具体的
- 目标是可度量的
- 目标是可实现的
- 目标是相关的
- 目标是有时间限制的

我建议大家在课程之前可以整理下自己的小目标，同时也可以将目标与课程的内容进行联系。被动学习的记忆是短暂的，很难持续停留，只有自己思考过程的知识才会被记住。所以在学习的过程中还是要自己独立思考，才会持续进步。

1.4 课程 SQ3R 学习法

任何知识不会自己被大脑记忆，它必须在实践过程中才会被具体的认知。人类是天生的阅读者，但最有效的学习方法就是实践。我希望大家带着自己的问题和观点去学习课程讲义，具体的推荐使用 SQ3R 学习方法。

SQ3R 学习法 (Survey, Question, Read, Recite, Review)：浏览、提问、阅读、复述和回顾。当你看到讲义中的定义、原理和例子时，你可能会想到“这些很显然很简单，我早就知道了”，由于讲义包含的内容比较基础，所以你大概都知道这些知识点，但相同的知识点也会用在不同的常见，会起到不同的作用。所以我希望你在阅读讲义的时候可以这样问自己：

- 讲义中的内容和“我”自己认知的有什么出入？
- 讲义中的内容有什么缺陷，是讲义没讲到，但“我”知道的。

- 讲义中的内容“我”能够给他人讲解清楚，让对方明白？
- 讲义中的内容与“我”自身的知识有什么关系？

课程的章节是按照知识点依次进行组织的，有基础的同学可以跳过前面的基础章节，直接阅读结构数据挖掘、计算机视觉任务和自然语言处理任务的章节。

1.5 课程基础

由于数据竞赛是一个交叉学科，所以需要有一定基础才能完全掌握讲义的全部内容。但也没有关系，因为我会尽量解释清楚每个知识点的含义，并给出一些参考链接供课后学习。

当然如果你有以下指示基础，会对学习课程有帮助：Python 语言、线性代数和概率论。课程中也会出现机器学习的一些名词概念，我会尽量在讲义中将其解释清楚，方便大家学习。

2 数据科学必知必会

在本章我会讲解数据科学的内容，以及数据科学/数据挖掘/机器学习/深度学习之间的关系；并从从业者/就业者两个角度来分享学习数据科学的收获和受益；最后我会讲解学习数据科学的路线和学习方法。

2.1 什么是数据科学？

1. 数据科学是什么？
2. 数据科学与机器学习有什么联系？
3. 数据科学能做什么？

在维基百科中数据科学定义如下：数据科学（Data Science）是一门利用数据学习知识的学科，其目标是通过从数据中提取出有价值的部分来生产数据产品²。数据科学结合了诸多领域中的理论和技术，包括应用数学、统计、模式识别、机器学习、数据可视化、数据仓库以及高性能计算。数据科学通过运用各种相关的数据来帮助非专业人士理解问题。

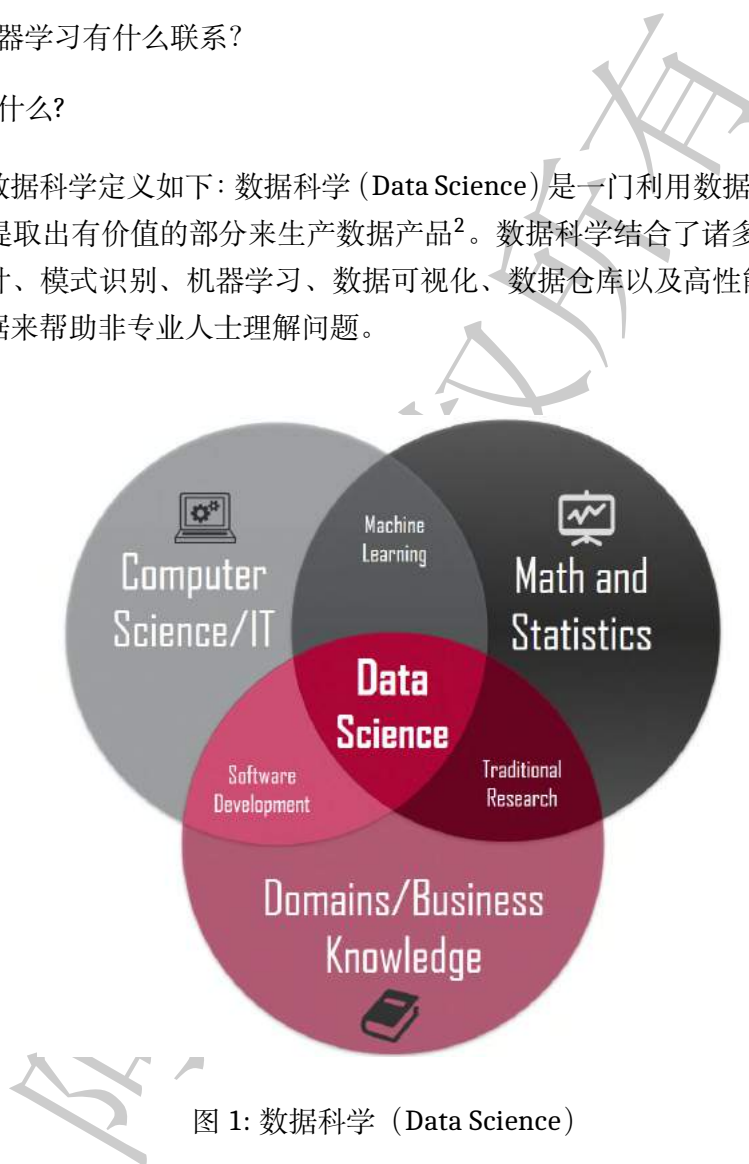


图 1: 数据科学（Data Science）

数据科学很是一门复杂性学科，其包含数据挖掘算法和机器学习算法。数据科学的工作流程是数据挖掘的流程，背后使用到的算法和优化方法来自机器学习。你可以理解为数据挖掘是数据科学的一种形式，机器学习是数据科学背后的思想，此外数据科学常用的技能还包括数据分析、数据可视化和数据存储等。

²http://en.wikipedia.org/wiki/Data_science

数据科学能做很多事情，只要有数据存在的地方就有其用武之地。为什么这样说呢，因为数据本身并不是随机存在的（其背后有分布规律），数据科学一方面可以对数据进行分析，并从流程上指导更好的实践。所以无论你的原本专业是什么，我都建议你了解一下数据科学，这是属于数据的时代。

2.2 为什么学习数据科学？

那么我们为什么要花费精力去学习数据科学呢？我想从以下三个角度分享下我个人的看法。

1. 从学业角度

从学业角度来说：数据科学能帮助你分析实验数据的规律，能够提供分析数据的工具，能够提供具体的解决思路。同时数据科学是一门实践性较强的学科，在这里有众多的实践场景，是验证你的专业能力的最好的方式。

2. 从就业角度

从就业角度来说：数据科学在国内外的大小公司都有对应岗位。比如数据挖掘岗、图像算法岗、广告算法岗和深度学习岗。学习数据科学能够扩宽就业面，有较好的就业机会。但数据算法岗和软件开发岗并不是完全对立的，因为数据科学具体的实践还是写代码，还是需要大家有良好的编程基础和计算机知识。

3. 从行业发展角度

从行业发展角度来说：数据科学是正在快速发展的学科，基本上每年都会有新方法、新工具和新应用的出现。数据科学在传统领域还有很多待解决的任务，发展潜力很大。

2.3 如何学习数据科学？

本次讲义是以数据科学以学习目标，具体以数据挖掘任务为研究对象。数据科学里面的知识点和设计的学科众多，接下来可以从以下三个方面讲解如何学习数据科学：

1. 数据挖掘/机器学习的定义

数据挖掘目标是从海量数据中挖掘出有用的信息，核心是找到变量之间的关系。机器学习目标是设计算法模型让其能够自动从经验中学习新知识。数据挖掘与机器学习相辅相成，数据挖掘很多任务都是用机器学习的方法完成的。你可以学习常用的数据挖掘算法，学习机器学习的优化技巧，学习公式的推导。

2. 数据挖掘的流程

你还可以学习数据挖掘的流程，具体包括以下几个步骤：数据分析、数据预处理、特征工程、构建模型和模型验证与部署，可以参考 TOPT 工具的流程。在这每个步骤中都有很多的细节，都是需要反复实践才能掌握的。

从原始数据开始到最终结果，都是数据的生命流程。在数据挖掘流程中各个步骤并不是单独存在的，步骤之间存在关联性。此外数据挖掘的流程也不是严格的有向图，步骤之间可以反复。

3. 数据挖掘的实践方法

数据挖掘是以数据为研究对象的任务，任务的核心点是数据。因此在实践的过程中最为重要的就是对数据的理解，并结合领域知识来对数据二次加工。理解数据，分析数据，实践。数据科学

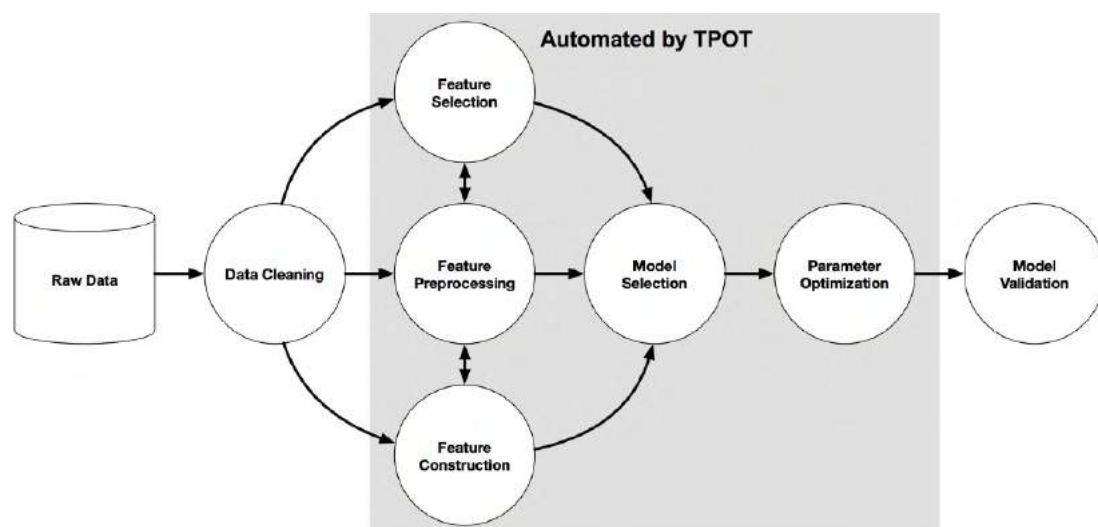


图 2: TPOP 工作流程

本身是一种编程活动，因此最终的成果还是用代码体现出来的，因此需要大家有良好的编程能力。

但数据科学应用和传统软件开发相比，两者还是有相似和相同的地方。首先两者都是属于软件开发的过程，由编码、调试和测试几个部分组成，其次两者都包括相似的数据结构和算法。

但两者还是存在很多差异的地方，首先两者的目的不一样：传统软件开发一般是实现某种功能，比如实现一个电商系统或者网络爬虫；数据科学应用侧重于数据相关的目的，比如数据的分析和数据的建模；其次两者的调试方法不一样，传统软件开发一般通过 **bug** 进行调试，只要没有 **bug** 可以实现目标的功能就够了，但数据科学的应用需要程序正确的运行和正确的结果。总而言之，数据科学不仅要程序能够正确运行，还要有“额外”的目标。

2.4 数据科学包含的知识点

数据科学是一门新鲜的学科，但其内容还是基础学科的内容。数据科学涉及到数据相关的方方面面，基础知识点可以从高等数学/微积分、线性代数、矩阵分析、概率论、信息论、最优化方法、图论和离散数学等课程中获得。数据科学包括的知识点主要是以概率论与数理统计为建模基础，进一步以线性代数和矩阵完整计算操作。

可以细分为以下几个知识点：

1. 微积分

- 函数极限
- 上确界与下确界
- 导数与偏导数
- 单调性与极值
- 函数的凹凸性
- 泰勒级数

- 牛顿-莱布尼兹公式
- Lipschitz 连续性
- Hessian 矩阵

2. 线性代数

- 线性空间与线性映射
- 行列式求解
- 特征值与特征向量
- 最小二乘法

3. 矩阵分析

- 矩阵运算
- 广义特征值
- 奇异值分解
- 矩阵求导
- 协方差矩阵

4. 概率论与数理统计

- 概率空间与事件
- 随机变量与概率公式
- 概率独立性
- 条件概率、联合概率与边缘概率
- 贝叶斯公式
- 大数定理与中心极限定理
- 参数估计

5. 信息论

6. 最优化方法

- 凸优化介绍
- 凸函数与凸集合
- 拉格朗日乘数法与 KKT 条件
- 常见的凸优化问题

7. 图论

- 图概念
- 常见的图
- 路径搜索问题
- 最大流问题
- 拉普拉斯矩阵

2.5 数据竞赛平台

本节介绍下国内外常见的数据竞赛平台，在这些平台上常年都有数据竞赛。同时这些竞赛平台上还会有很多参赛选手的分享，非常适合学习，推荐大家有时间一定要去参加一下。

表 1: 数据竞赛平台

平台名	链接	介绍
Kaggle	www.kaggle.com	全球最大的竞赛比赛平台，竞赛机制完备
DrivenData	www.drivendata.org	较为成熟的平台，以图像和视频比赛为主
Colalab	competitions.codalab.org	举办大型学术类型比赛
CrowdAI	www.crowdai.org	举办大型学术类型比赛
天池	tianchi.aliyun.com	阿里旗下，知名度最高的中文竞赛平台
点石	dianshi.baidu.com/competition	百度旗下的数据竞赛平台
JData	jdata.jd.com	京东旗下的数据竞赛平台
DataCastle	www.pkbigdata.com	电科大背景，国内较为成熟的平台
DataFountain	www.datafountain.cn	CCF 背景，每年举办 CCF 数据挖掘竞赛
Biendata	biendata.com	清华学术背景，平台以学术比赛为主
科赛	www.kesci.com	国内机制较为完善的平台

近年来数据竞赛也非常火，国内外的互联网企业每年都会举办各种数据竞赛。一方面数据竞赛可以给举办公司做广告，另一方面举办公司也希望数据竞赛中收获参赛选手的解决方案。最典型的例子是 2006 年 NETFLIX 举办的推荐系统比赛，第一个能把现有推荐系统的准确率提高 10% 的参赛队伍将获得一百万美元的奖金。而最终第一名的解决方案完全优胜于 NETFLIX 公司自己的算法。

参加数据竞赛非常考验动手能力，参赛选手需要对数据进行深入理解，并根据业务背景进行特征工程。每一场数据竞赛是有具体的业务场景的，是工业界或者学术界的具体问题，都是非常有价值的。对于参赛者来说，参加比赛能够学习技术并证明自己的能力，也可以获得较好的求职/升学 offer。所以推荐大家有时间一定要去参加一场数据竞赛。

2.6 竞赛实例讲解

接下来介绍下竞赛的相关知识点，首先按照赛题的任务可以将赛题类型分为三类：

- **分类赛题**：比赛的标签是类别，任务是分类问题。例如预测用户是否违，图像分类；
- **回归赛题**：比赛的标签是数值，任务是回归问题。例如预测用户的贷款金额，PM2.5 预测；
- **时序赛题**：比赛的标签与时间相关，任务是时序问题。例如商铺销量预测，汽车流量预测；

赛题也可以根据数据类型分为两类：

- **结构化数据**：数据已表格形式进行表示，例如表格数据；
- **非结构化数据**：数据已非结构化进行表示，例如文本数据或者图像数据；

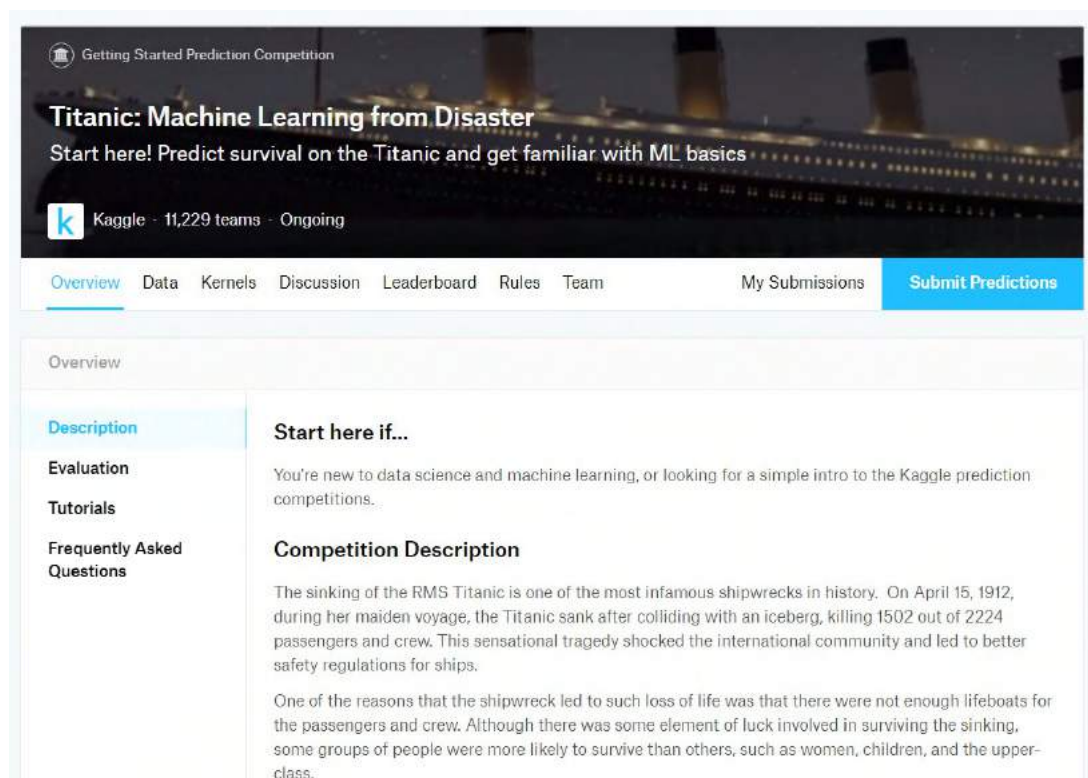


图 3: Titanic: Machine Learning from Disaster

赛题也可以根据业务场景进行分类：风控类型问题、CTR 类型问题和用户行为预测等场景。当你拿到赛题后，一定要把出题方给定的信息进行仔细阅读，弄清楚赛题的背景、任务、数据、评测方式、时间点等信息。我也推荐大家有空就去 Kaggle 上多参加一些比赛，非常锻炼能力，同时也能够学到很多。

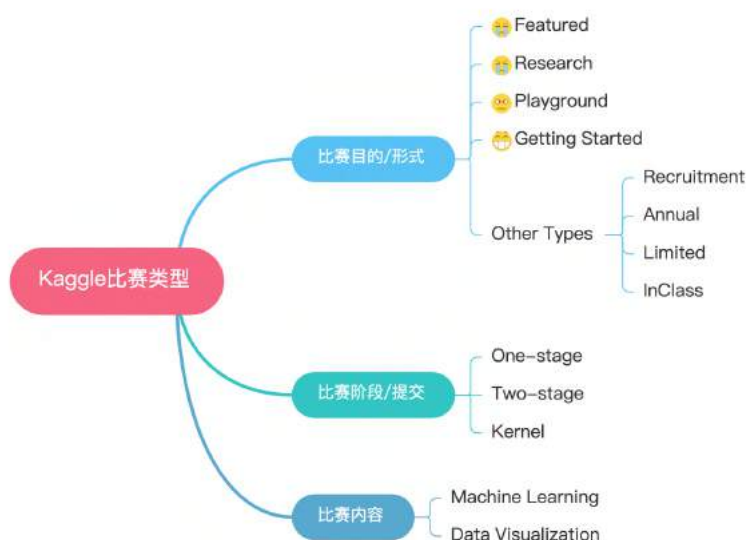


图 4: Kaggle 比赛类型

Kaggle 上每个比赛都会包括如下的页面：

1. Overview 页面: 对比赛的背景任务和评测方式相关的介绍，还包括比赛时间赛程信息；

2. Data 页面: 对比赛数据信息的介绍;
3. Kernels 页面: 对比赛分享的代码内核, 可以是 R 或者 Python 两种环境的。Kernels 还提供了免费的计算资源 (还有 GPU 哦), 对选手非常友好;
4. Discussion 页面: 是比赛相关的帖子分享;
5. Leaderboard 页面: 是比赛得分的排行榜;
6. Rules 页面: 比赛的规则和注意事项;
7. Team 页面是比赛的组队信息;

每个比赛的 Kernels 和 Discussion 部分都是广大参赛选手进行分享的地方, 会分享赛题的方方面面, Kaggle 的魅力就在于此, 每个比赛的 Kernels 和 Discussion 部分都会让人醍醐灌顶。



图 5: Kaggle 解题流程

2.6.1 Rong360-用户贷款风险预测

https://www.pkbigdata.com/common/cmpt/用户贷款风险预测_竞赛信息.html

赛题: 根据用户的基本属性、银行流水记录、用户浏览行为、信用卡账单记录、放款时间, 以及这些顾客是否发生逾期行为的记录。

目标: 风控背景的二分类问题, 预测用户是否违约;

思路: 根据不同的表, 提取用户不同的特征;

难点: 用户 ID 存在 leak, 且放款时间存在特殊分布; 需要有较好的交叉特征方法;

2.6.2 Planet: Understanding the Amazon from Space

<https://www.kaggle.com/c/planet-understanding-the-amazon-from-space>

赛题: 遥感图像, 判断地表的天气和地表覆盖物;

目标: 对天气和地表进行多分类;

思路: 使用预训练模型 (ResNet/Desnet/Inception...), 微调 CNN 模型;

难点: 类别分布差异大, 且训练需要 4*1080ti 的计算能力;

2.7 本章小结

本章讲解了数据科学的基础知识以及数据竞赛的介绍, 在之后的章节中我们将会以此为基础介绍数据挖掘的流程, 并介绍如何解决一个数据竞赛问题。



图 6: Planet 图像类别

3 机器学习基础

本章的基本内容是机器学习 (Machine Learning) 的基础, 机器学习包含的知识点非常多, 同时也可以从多个角度对机器学习的相关概念进行学习。机器学习也包括众多种类的机器学习算法, 每一类机器学习算法都包含很多实现细节。在具体的实践过程中, 我们需要关注不仅是算法细节, 还要更加关注整个流程。在实践的过程中需要知道需要做什么, 下一步应该做什么, 继续怎么改进。

你不仅要知道有哪些机器学习算法, 还应该知道每种算法的缺点和适用场景。

在本章中, 首先会对机器学习的一些基础概念进行讲解, 并介绍了常见的误差度量方法以及常见的机器学习算法。在讲解具体机器学习算法的部分, 本书侧重于每种机器学习算法的优缺点, 以及偏好的应用场景。由于篇幅原因, 在章节内容中并不会展开公式推导的细节, 如果读者感兴趣可以查阅相应的教材。

3.1 误差与过拟合

对于分类任务我们可以用分类错误率来衡量模型的性能, 具体来说模型在训练集上的误差成为训练误差, 模型在新样本 (测试集) 上的误差成为泛化误差, 评估机器学习应该使用泛化误差进行评价。

在模型的训练过程中, 模型只能利用训练数据来进行训练, 模型并不能接触到测试集上的样本。因此模型如果将训练集学的过好, 模型就会记住训练样本的细节, 导致模型在测试集的泛化效果较差, 这种现象称为过拟合 (Overfitting)。与过拟合相对应的是欠拟合 (Underfitting), 即模型在训练集上的拟合效果较差。

- **过拟合**: 模型在训练集误差较低, 但在测试集上误差较高;
- **欠拟合**: 模型在训练集误差较高, 还没有完全拟合训练集;

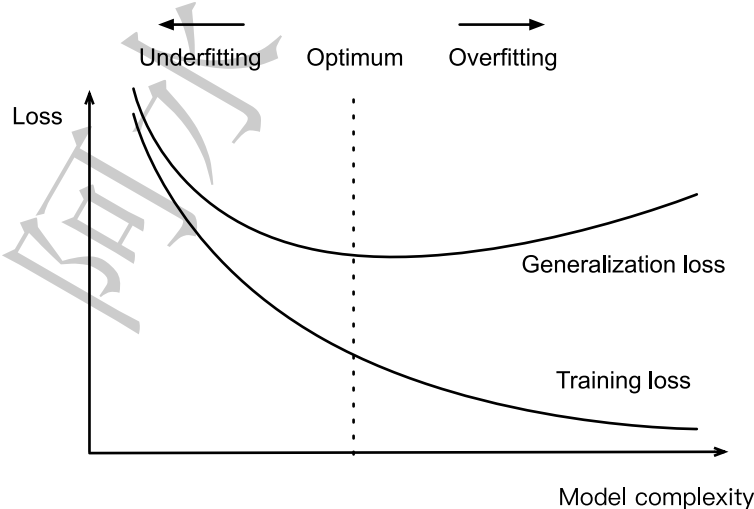


图 7: 欠拟合与过拟合

导致模型过拟合的情况有很多种原因, 其中最为常见的情况是模型复杂度 (Model Complexity) 太高, 导致模型学习到了训练数据的方方面面, 学习到了一些细枝末节的规律。欠拟合由模型复杂度较

高或模型未训练完全导致的，解决方案很简单：增加模型的复杂度或者增加模型的训练轮数。

3.2 模型评价方法

机器学习任务可分为：分类、回归和排序三种。其中每种任务的侧重点不同，因此可用不同的评价函数进行度量。对于有监督任务，误差通过样本标签与模型预测的标签进行对比；对于无监督任务，则需要具体的任务定义具体指标。

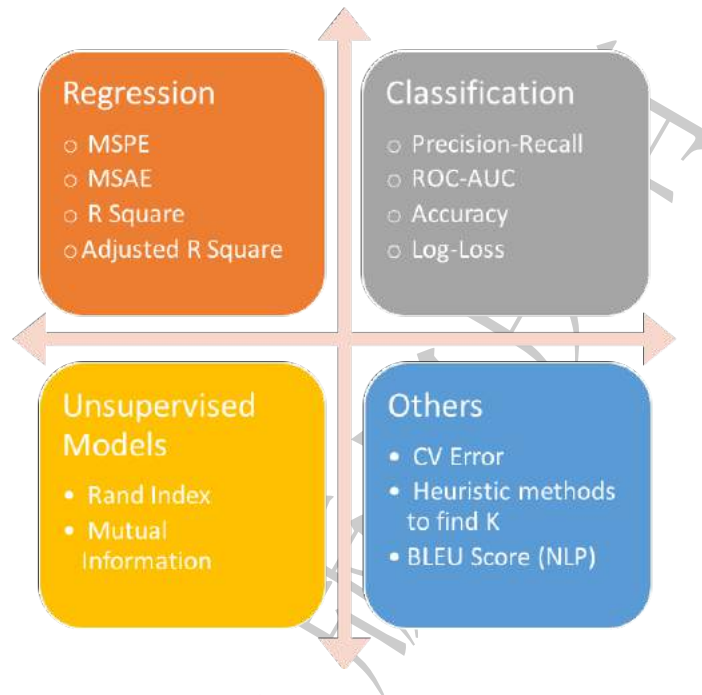


图 8: 常见的评价方法

混淆矩阵 (Confusion Matrix)，针对二元分类问题，将实例分为正类或者负类，会出现四种情况：

- (1) 实例本身为正类，被预测为正类，即真正类 (TP)；
- (2) 实例本身为正类，被预测为负类，即假负类 (FN)；
- (3) 实例本身为负类，被预测为正类，即假正类 (FP)；
- (4) 实例本身为负类，被预测为负类，即真负类 (TN)；

1. 准确率

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

准确率在类别极度不平衡问题上，作为评级指标不是十分合适。例如，1000 个样本中，990 个正例，10 个负例，分类器预测准确率为 90%，而全将样本作为正例都有 99% 的准确率。

2. 查准率和召回率

查准率 (Precision)：模型预测为正例数据占预测为正例数据的比例。

$$Precision = \frac{TP}{TP + FP}$$

		Condition (as determined by "Gold standard")		
		Condition Positive	Condition Negative	
Test Outcome	Test Outcome Positive	True Positive	False Positive (Type I error)	Positive predictive value = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Test Outcome Positive}}$
	Test Outcome Negative	False Negative (Type II error)	True Negative	Negative predictive value = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Test Outcome Negative}}$
		Sensitivity = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Condition Positive}}$	Specificity = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Condition Negative}}$	Accuracy = $\frac{\text{True positive} + \text{True negative}}{\text{All}}$

图 9: 混淆矩阵

召回率 (Recall): 预测为正例的数据占实际为正例数据的比例

$$Recall = \frac{TP}{TP + FN}$$

3. F1 值 (F1 Score)

F1 值同时考虑了准确率和召回率:

$$F1 = \frac{2}{1/P + 1/R} = \frac{2 * P * R}{P + R}$$

$$F_{\beta} = \frac{1 + \beta^2 * P * R}{\beta^2 * P + R}$$

4. ROC 和 AUC

ROC 是由点 (TPR, FPR) 组成的曲线, AUC 就是 ROC 曲线下面积, AUC 越大越好。

5. MAP (Mean average precision) MAP 全称 Mean Average Precision, 表示平均正确率, 是每个类别准确率的平均结果。假设有两个主题, 主题 1 有 4 个相关网页, 主题 2 有 5 个相关网页。某系统对于主题 1 检索出 4 个相关网页, 其 rank 分别为 1, 2, 4, 7; 对于主题 2 检索出 3 个相关网页, 其 rank 分别为 1, 3, 5。对于主题 1, 平均准确率为 $(1/1 + 2/2 + 3/4 + 4/7)/4 = 0.83$ 。对于主题 2, 平均准确率为 $(1/1 + 2/3 + 3/5 + 0 + 0)/5 = 0.45$ 。则 $MAP = (0.83 + 0.45)/2 = 0.64$ 。

6. NDCG (Normalized Discounted Cumulative Gain)

7. MRR (Mean reciprocal rank) MRR 是对搜索算法进行评价的机制, 指多个查询语句的排名倒数的均值。

8. 均方根误差 (Root Mean Square Error, RMSE)

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

9. 平均绝对误差 (Mean Absolute Error, MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

10. (Mean Absolute Percentage Error, MAPE)

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

11. R Squared (R^2)

$$\hat{R}^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

3.3 偏差与方差

学习算法的预测误差，或者说泛化误差 (Generalization error) 可以分解为三个部分：偏差 (Bias)，方差 (Variance) 和噪声 (Noise)。在估计学习算法性能的过程中，我们主要关注偏差与方差，因为噪声属于不可约减的误差。

$$Err(x) = \mathbb{E}_D \left[(f(x; D) - \bar{f}(x))^2 \right] + (\bar{f}(x) - y)^2 + \mathbb{E}_D \left[(y_D - y)^2 \right]$$

▶ variance
 ▶ bias²
 ▶ noise

1. 偏差：度量了算法预测结果与真实结果的偏离，刻画了算法的拟合能力；
2. 方差：度量了同样大小的训练集的变动所导致的学习性能的变化，即数据扰动所造成的影响；
3. 噪声：表达了在当前任务上任何算法能达到的泛化误差的下界，即；刻画了学习问题本身的难度；

方差与偏差的分解表明，模型的泛化性能是由模型的学习能力、数据量以及数据噪音所决定的。同时偏差与方差是有冲突的，但模型在欠拟合状态时，模型对训练集的拟合程度不够，数据的扰动不足以影响模型；模型在过拟合情况下，模型拟合能力很强，模型也会学习到数据的扰动。

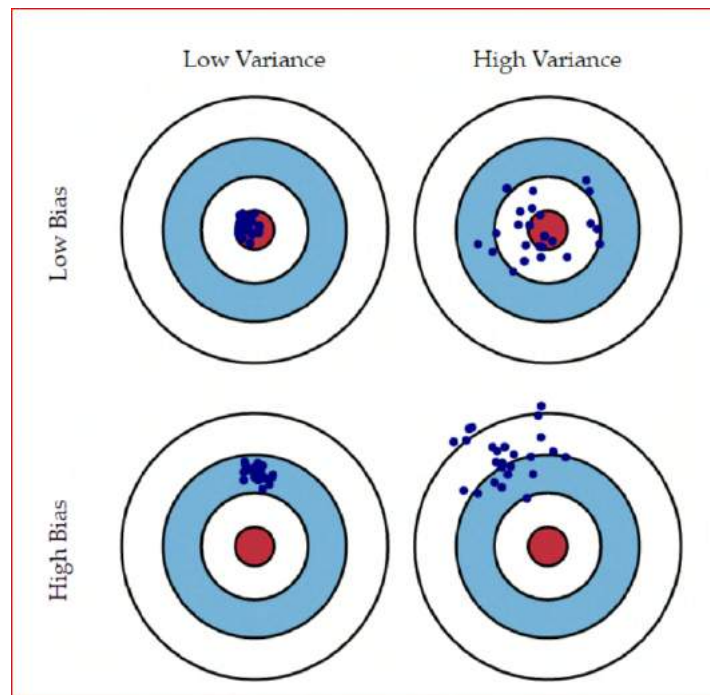


图 10: 方差与偏差

3.4 线性模型

线性模型（Linear Model）顾名思义，模型是线性的模型。

$$f(x) = w_1x_1 + w_2x_2 + \dots + w_dx_d + b$$

$$f(x) = w^T x + b$$

线性模型虽然简单，但具备很多优点。首先线性模型的解释性强，模型的参数可以表示特征的重要性；其次线性模型转换为非线性模型，只需要引入高维空间即可；最后线性模型非常适合分布式训练。

线性模型中常见的模型包括线性回归（Linear Regression）和逻辑回归（Logistic Regression, LR），前者利用线性模型完成回归任务，后者利用线性模型完成分类任务。

线性模型虽然有较多优点，但在应用的场景还是有很多缺点：首选线性模型能力有限，只能完成模型的线性组合；其次线性模型对特征进行加权求和，因此对特征的量纲有偏好，需要提权对输入进行预处理。

3.5 树模型

树模型（Tree Model）是一类常见的机器学习算法，是以树结构来进行决策的，也是比较类似人类决策的过程。一颗决策树包含一个根节点、若干个内部节点和若干个叶子节点；每个叶子阶段对应决策结果，非叶子节点对应一个决策情况。

决策树的构建是一个递归的过程，不断地根据当前样本的属性分布进行划分，直到无法继续分裂为止。决策树是根据最优属性进行划分的，一般可以用信息熵、信息增益和基尼指数等指标进行衡量。

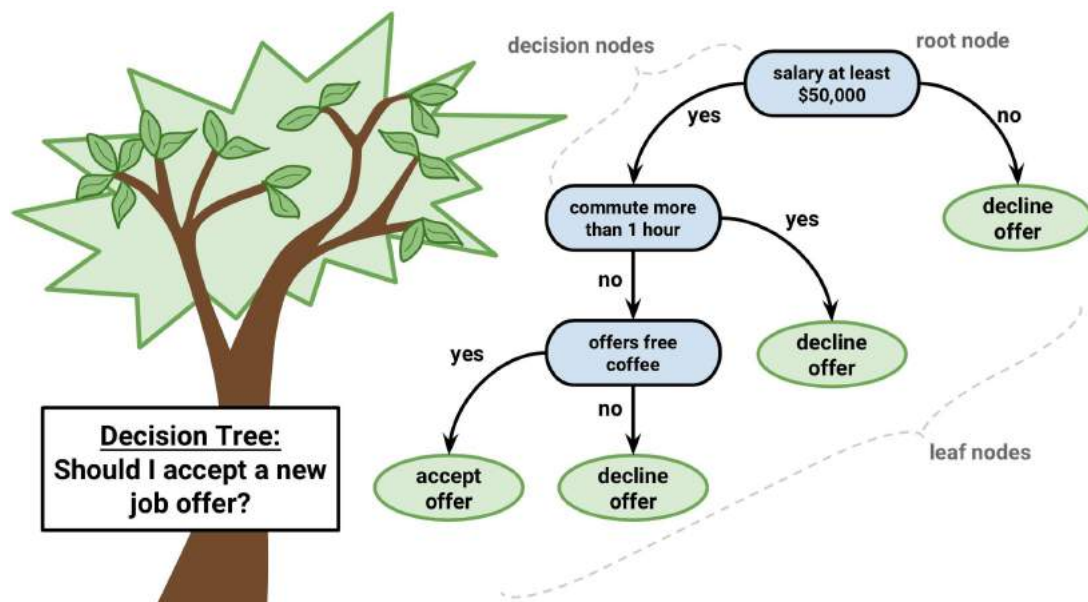


图 11: 树模型

由于决策树每次只使用单个属性进行决策，因此决策树对量纲并不敏感。同时树模型对缺失值也比较友好，可以将缺失值视为一个特殊的取值。决策树在训练过程中可以加入节点随机选择、Boosting 和 Bagging 等技术，进一步增加树模型的拟合能力。

3.6 KNN 模型

KNN 模型（K 近邻模型，K-Nearest Neighbor）是一种常见的机器学习算法，算法思路非常简单。给定测试样本，样本的标签取决于最近 K 个邻居样本的取值情况。对于分类任务，KNN 选择最近的 K 个邻居完成投票或者结果平均；对于回归任务，KNN 选择最近的 K 个邻居完成样本的结果平均。

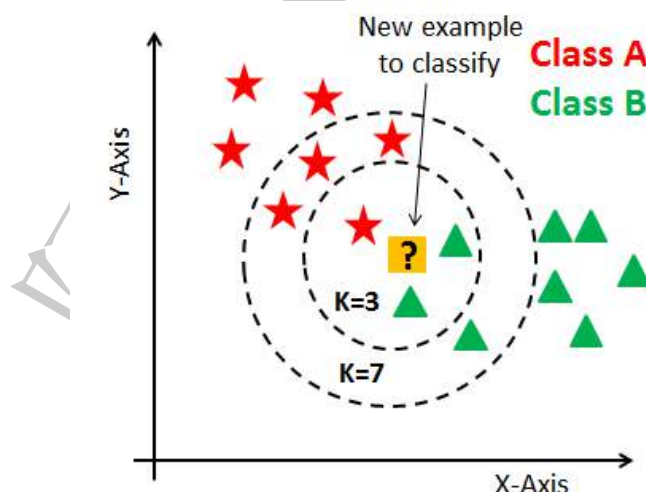


图 12: KNN 模型

KNN 模型没有训练的过程，只需要在测试的阶段完成对应的计算操作。所以 KNN 模型所有的中心流程就是如何完成样本的距离计算，一般情况下可以直接用欧氏距离完成，也可以自定义距离函数。

Sklearn 中聚类方法

在 Sklearn 中定义了众多聚类算法和 KNN 的实现，用起来非常方便。

- `sklearn.neighborsa`: KNN 算法的相关函数，如 KNN 分类器和 KNN 回归函数；
- `sklearn.clusterb`: 常用的聚类算法，主要区别在于距离函数定义和聚类过程；
- `sklearn.manifoldc`: 流形学习的相关算法；

^a<https://scikit-learn.org/stable/modules/classes.html#modulesklearn.neighbors>

^b<https://scikit-learn.org/stable/modules/classes.html#modulesklearn.cluster>

^c<https://scikit-learn.org/stable/modules/classes.html#modulesklearn.manifold>

3.7 神经网络

神经网络 (Neural Networks) 最早起源于研究者对神经元的认识，进而发展成为感知机模型。神经网络中最基础的元素是神经元，神经元之间相互连接，可以相互发送信号。

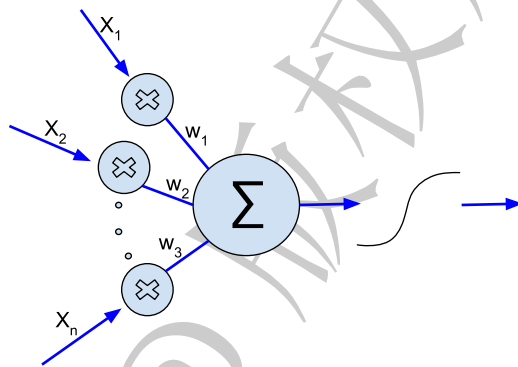


图 13: 神经元模型

但神经元被激活后，它就会被激活，向后序的神经元发送信号。在神经元模型中神经元会受到前序神经元的信号输出，并将信号经激活函数处理后输出，典型的神经元模型是 Sigmoid 函数。将神经元按照一定规律组合后，就成为神经网络。神经网络通常使用误差反向传播进行优化，并按照批量样本多次进行梯度反传。

神经网络拥有强大的拟合能力，只包含单隐含层的神经网络可以拟合任意复杂度的函数。神经网络的参数越多、层数越多，模型的拟合能力越强。随着数据量不断增长，GPU 设备加快了神经元的并行训练，深度学习逐渐大放异彩。特别是适用于计算机视觉任务的卷积神经网络 (Convolutional Neural Network, CNN) 横扫了视觉类的所有任务，在图像分类、图像检索、物体检测和图像分割等领域，CNN 都在精度上超过了人类水平。

3.8 本章小节

本章讲解了机器学习的评价函数和常用的机器学习模型，需要注意的是学习机器学习一定要理论加实践，一定要明白各个算法的原理和应用场景，一定要知道如何实践这些模型。

4 常见的树模型

树模型是应用非常广泛的机器学习模型，决策树本质做的是做一系列条件决策过程。

4.1 Isolation Forest

Isolation Forest（孤立森林）是由周志华等人在 2007 年提出的用于异常检测的树模型。

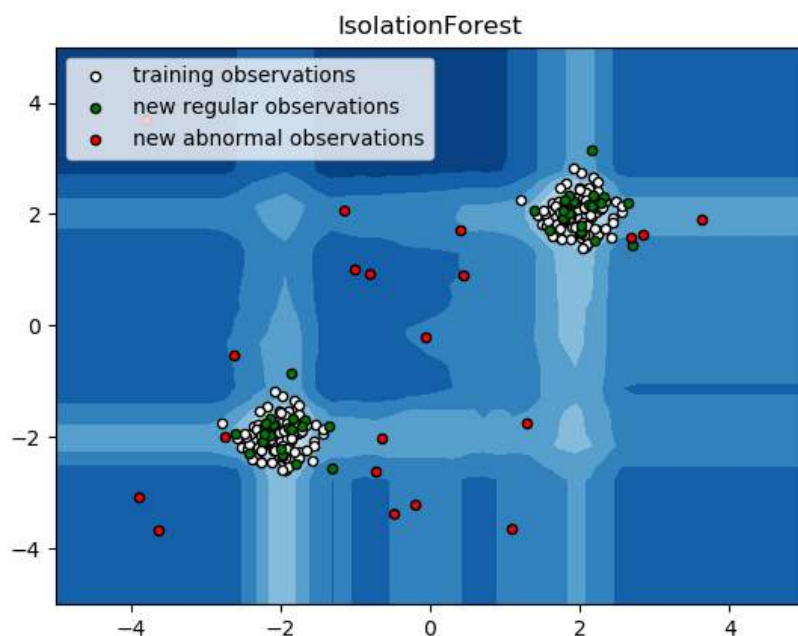


图 14: Isolation Forest 例子

4.2 随机森林

TODO

4.3 GBDT

TODO

4.4 XGBoost

TODO

4.5 LightGBM

TODO

4.6 CatBoost

TODO

4.7 本章小结

TODO

阿水@版权所有

5 深度学习

深度学习（Deep Learning）是机器学习的分支，是一种以人工神经网络为架构，对数据进行表征学习的算法。深度学习本质是多层的神经网络，具有较强的特征学习能力。深度学习模型学习的学习能力是分布式表示的，并利用了分层抽象的思路，高层次的概念从低层次的概念学习得到。基于这种想法，早期深度学习的每层常使用贪心算法逐层构建³。逐层训练大大减少了训练开销，也是“预训练+微调”思路的源头。

神经网络的隐含层越多、模型参数越多，模型拟合能力更强，同时训练的难度也会增加。减少模型参数的方法有两种：逐层训练和权重共享。权重共享思路是现有深度学习的基础，它大大减少了模型的参数。

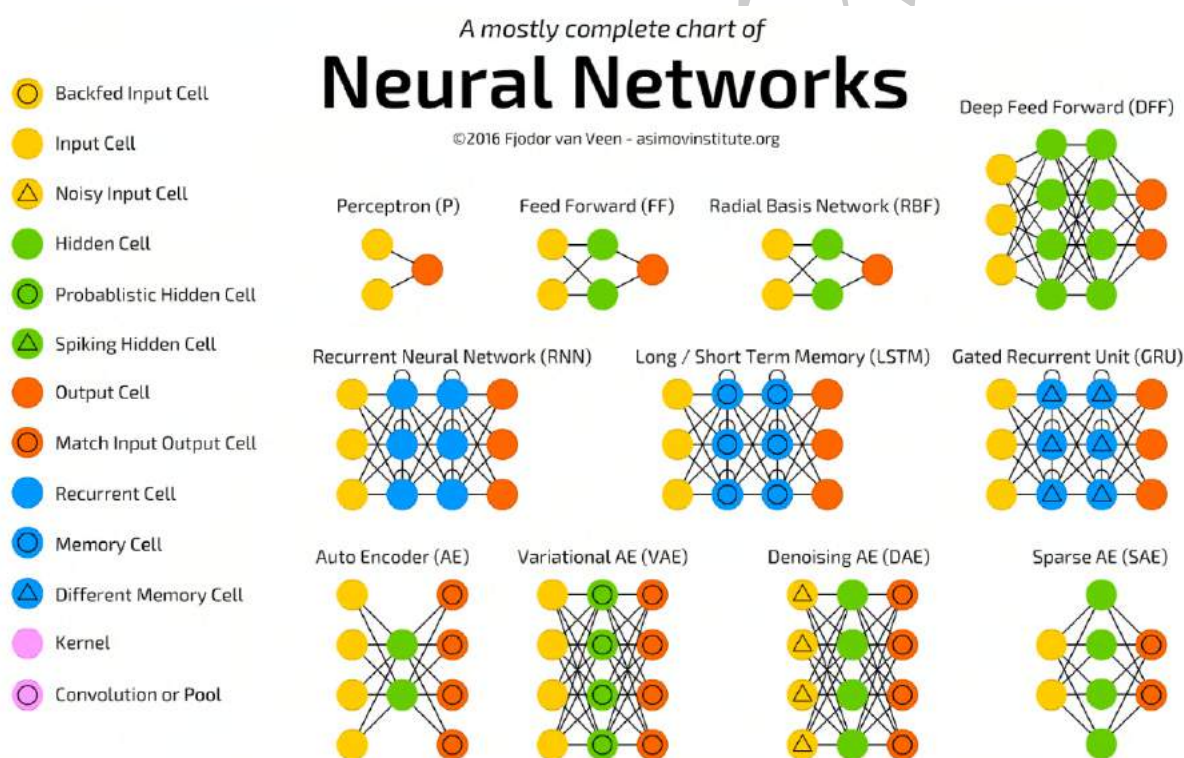


图 15: Deep Learning Model Zoos

5.1 基本原理

深度学习作为机器学习中的一个分支，基本定义（偏差与方差、误差与过拟合等）与传统的机器学习类似，但深度学习还是有很多独有的特点。首先深度学习并不需要很多的人工特征工程，利用其强大的建模能力来代替，整个训练的过程是端到端的过程（End-to-End）；其次深度学习模型参数居多，训练过程中需要大量的训练样本。

比如在图像分类任务中，传统的方法大都是先使用提取图像特征，然后利用机器学习模型来完成对应的任务。这种方法比较繁琐，需要针对不同的做不同的特征工程；且提取特征的步骤和模型训练

³<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8694781>

的步骤是拆开的。深度学习利用网络强大的特征表征能力，能够提取多层的高层特征，并将特征空间与标签空间直接进行映射。

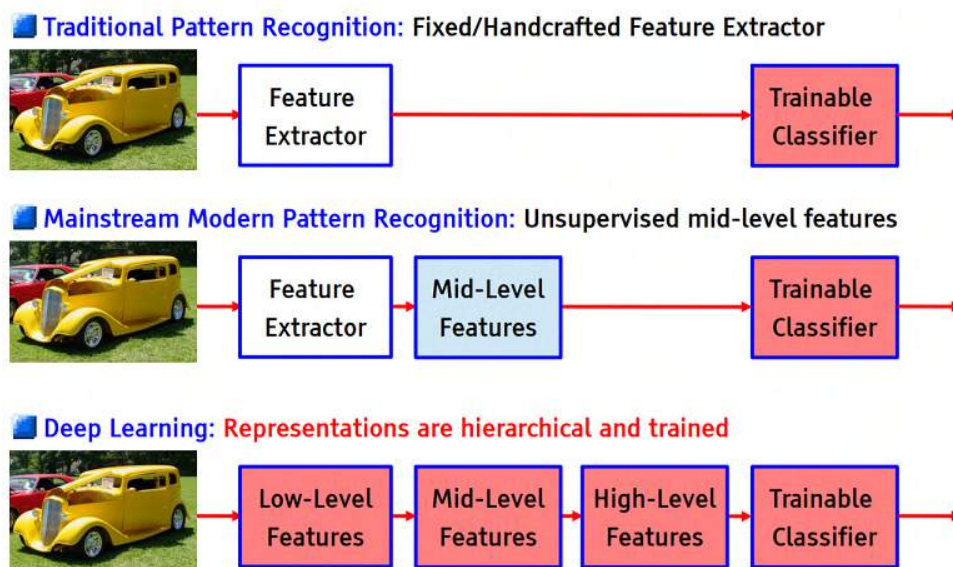


图 16: 深度学习流程

深度学习多层网络可以将数据进行多层映射，进行逐层学习。在卷积神经网络网络中底层网络可以学习到图像的底层信息，如边缘和角点，高层网络可以学习到图像的直接和复杂形状。

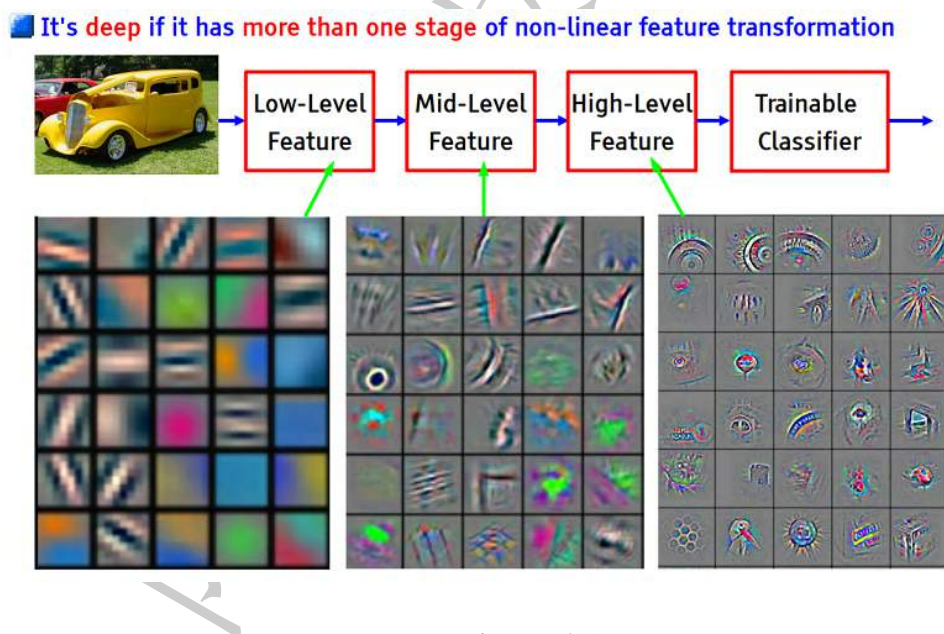


图 17: 深度学习表征

5.1.1 全连接网络

全连接网络（Full Connected, FC）是基础的深度学习模型，它的每一个神经元把前一层所有神经元的输出作为输入，其输出又会给下一层的每一个神经元作为输入，相邻层的每个神经元都有“连接”。

全连接的核心操作是矩阵乘法，本质上是把一个特征空间线性变换到另一个特征空间。全连接网络如果没有隐含层，就变成了全连接层。全连接层对输入的维度敏感，同时也包含了较多的冗余参数，

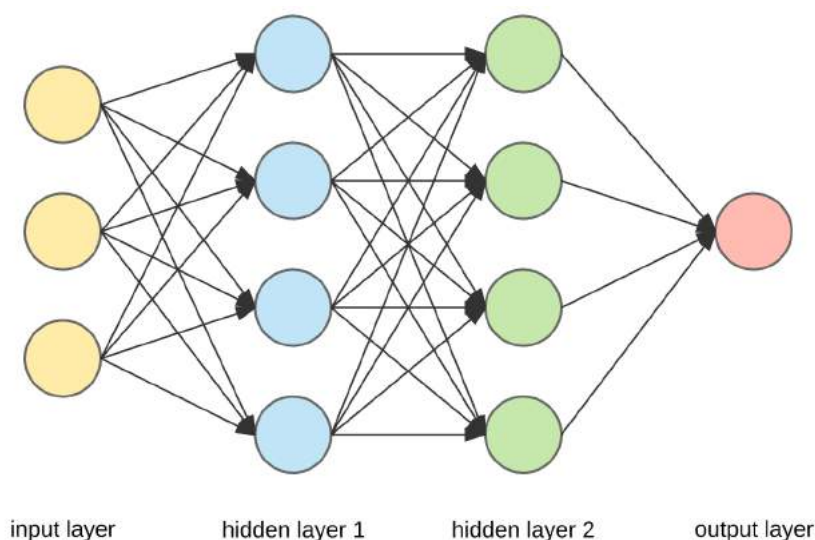


图 18: 全连接网络

也很难提取局部信息和时序信息。

5.1.2 深度学习的正则化

在机器学习中，正则化是指基于增强模型泛化能力的先验知识。深度学习拥有较多的参数，如果没有正则化技术，模型很容易陷入过拟合状态。深度学习的正则化技术包括参数正则化、Dropout、Early Stop、数据增强、梯度裁剪和标签平滑等技术。

1. Dropout

Dropout 是指在深度学习网络的训练过程中，对于每层网络单元，可以按照一定的概率将其暂时从网络中丢弃。因此每次前向传播的过程中，被丢弃的节点是随机宣告的。对于随机梯度下降来说，由于是随机丢弃，故而每一个 mini-batch 都在训练不同的网络。

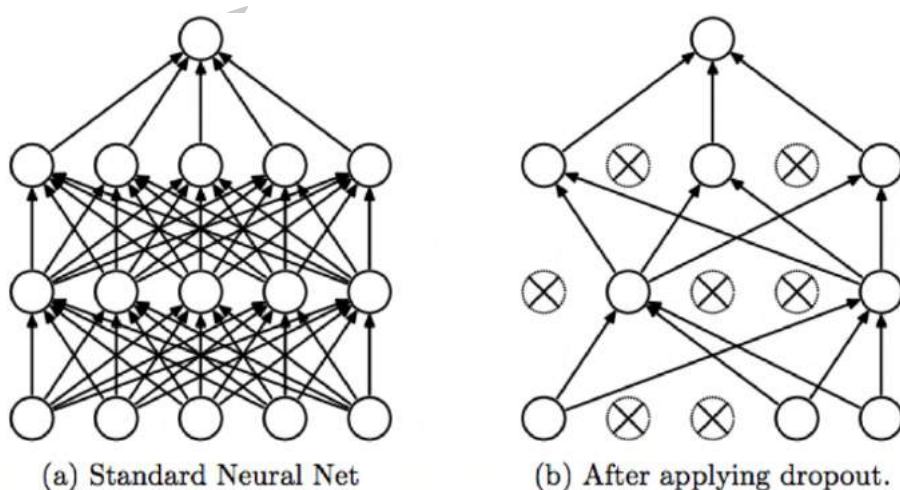


图 19: Dropout

在预测阶段，Dropout 应该关闭。Dropout 给神经网络带来了随机性，Dropout 关闭的情况下等同于多个模型进行了集成。

2. 数据增强 (Data Augmentation)

数据增强是用来扩充数据集，可以进一步减缓模型过拟合的情况。在图像分类任务中，图像的翻转、旋转、颜色改变、边缘处理等操作都不会改变图像的标签。在其他任务中，也可以使用类似的数据增强操作。

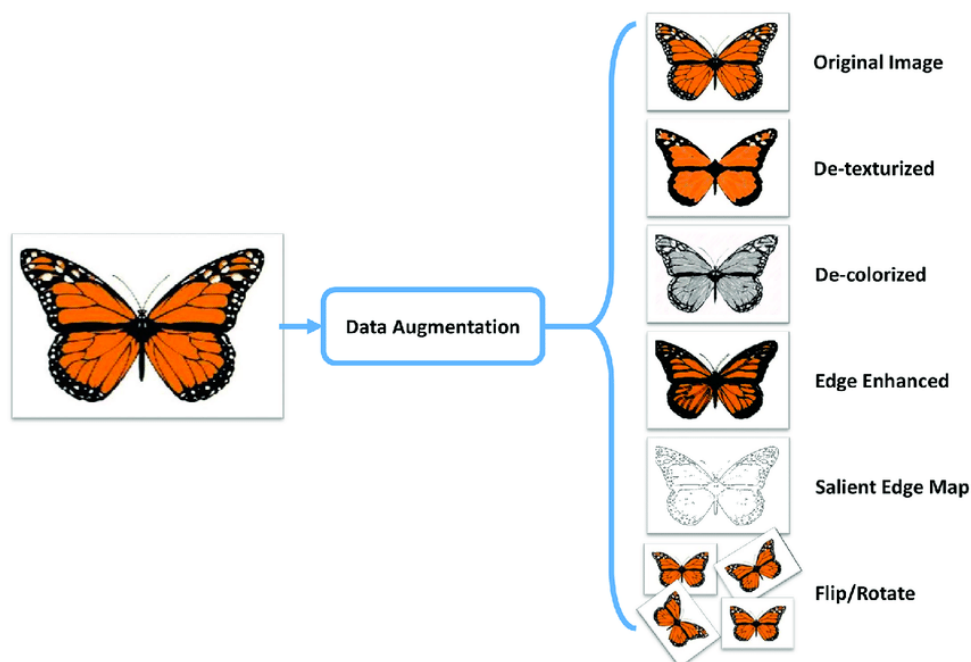


图 20: 数据增强

数据增强一般在训练过程中动态进行，因此每次迭代过程中都会有不同的图像进行训练。在预测阶段，也可以对图像进行数据扩增，然后对扩增图像的结果进行平均操作。

5.1.3 深度学习的优化

使用整个训练的优化算法成为批量 (Batch) 梯度算法，这种方式会在一个批量中处理所有的训练样本。每次只使用一个样本的优化算法称为随机 (Stochastic) 或在线 (Online) 算法，这种方式会每次从数据流中抽取样本。现有的深度学习大都使用小批量 (Mini-Batch) 方法，这种方法使用一个以上而又不是所有的训练样本，这里的样本个数又称为 Batch Size，是深度学习的一个超参数。

5.2 卷积神经网络

卷积神经网络与全连接网络非常相似：它们都是由神经元组成，都具有学习能力的权重和偏差。神经元都得到一些输入数据，进行内积运算后再进行激活函数运算。卷积神经网络的训练过程也是端到端的过程：输入是原始的图像像素，输出是不同类别的评分。

卷积神经网络一般由卷积层、激活函数、池化层和全连接层组成。卷积层由众多的卷积核组成，卷积核会以滑动的方式对输入范围的输入进行内积求和操作。卷积是一种局部操作，通过一定大小的卷积核作用于局部图像区域获得图像的局部信息。卷积层本质就是图像滤波器，而滤波的参数可以在训

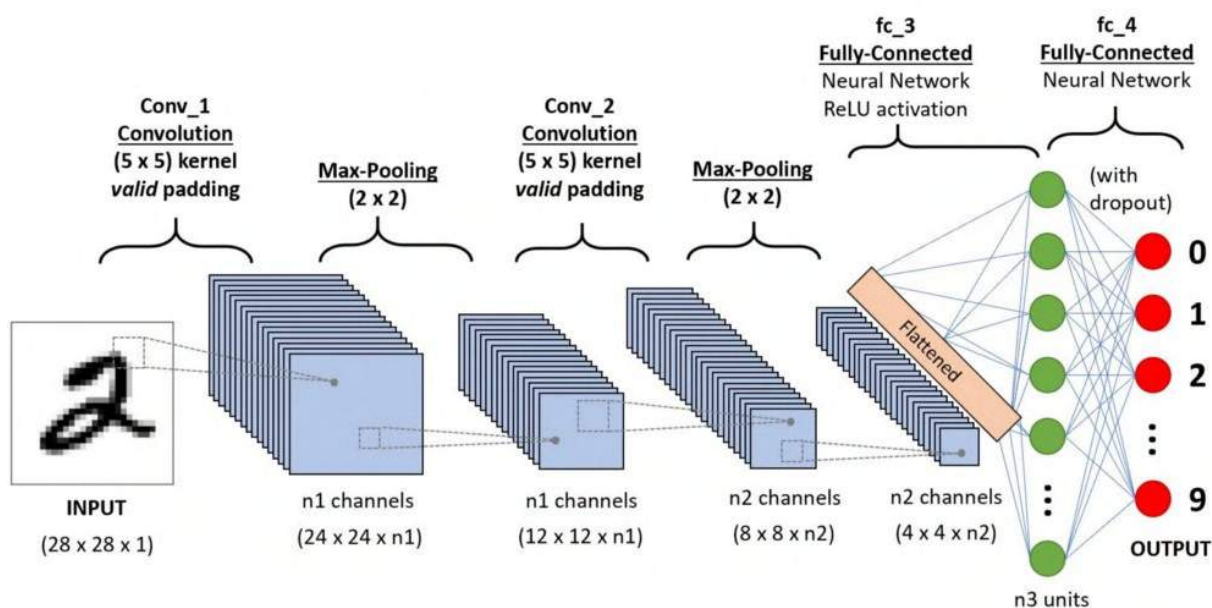


图 21: 卷积神经网络

练中动态调整，除了可以学习到边缘滤波器，还可以学习到检测形状、颜色的信息。

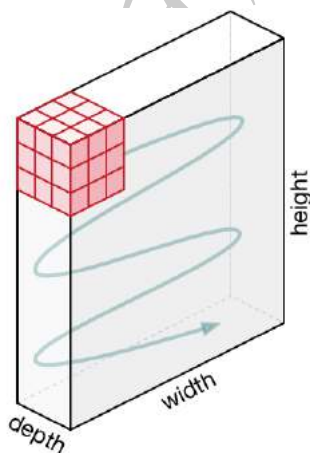


图 22: 卷积核

池化层的作用是进行特征降维和特征聚合的操作。卷积层一般包含多个卷积核，因此卷积操作会提取多维的数据，但并不是每个维度的特征是有效的。池化层可以提取每个通道局部的最大值和平均值等局部特征，而这些特征往往具有较好的尺度不变性。

5.3 循环神经网络

循环神经网络 (Recurrent Neural Networks, RNN) 是一类具有短期记忆能力的神经网络。在循环神经网络中，神经元不但可以接受其它神经元的信息，也可以接受自身的信息，形成具有环路的网络结构。

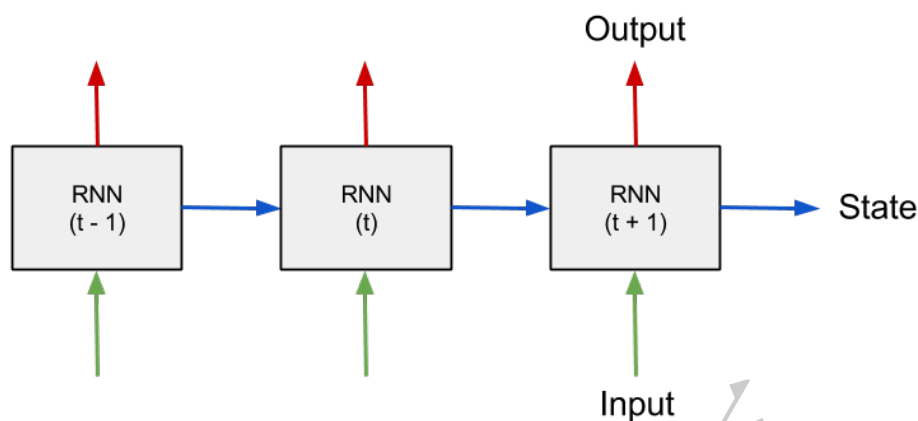


图 23: RNN

RNN 主要用于序列数据的分类问题：输入为序列，输出为类别。比如在文本分类中，输入数据为单词的序列，输出为该文本的类别。

5.4 自编码器

自编码器（Auto-Encoder）是一种数据的压缩算法，其中数据的编码器和解码器从数据中自动学习的。在大部分提到自编码器的场合，编码器和解码器都由神经网络实现。自编码器是一种无监督学习算法，其中编码器将输入进行压缩，解码器将压缩特征进行恢复。

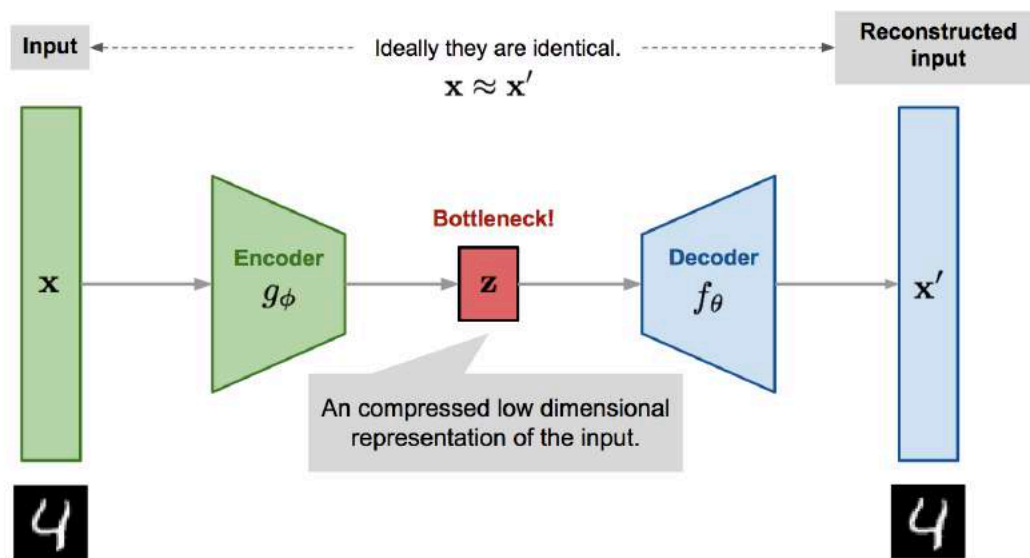


图 24: 深度学习训练流程

自编码器可学习到数据的内在维度，因此可用于特征降维和数据降噪。

5.5 深度学习的实践论

深度学习原理少但实践性非常强，基本上很多的模型的验证只能通过训练来完成。同时深度学习有众多的网络结构和超参数，因此需要反复尝试。训练深度学习模型需要 GPU 的硬件支持，也需要较

多的训练时间，如何有效的训练深度学习模型逐渐成为了一门学问。

深度学习有众多的训练技巧^{4 5}，本节挑选了常见的一些技巧来讲解，但针对具体问题还是要具体分析。炼丹有风险，也有很大的随机性存在。

1. 数据扩增很重要，会直接影响模型的泛化性能；同时不同的任务数据扩增的方式可能存在差异，所以要具体任务具体定制数据扩增的方式。
2. Dropout 只在训练阶段开启，在测试阶段要保持关闭，否则模型的预测结构会带有一定随机性。
3. 在训练过程中要观察误差曲线以及学习率，Early Stop 能够减缓过拟合的情况；

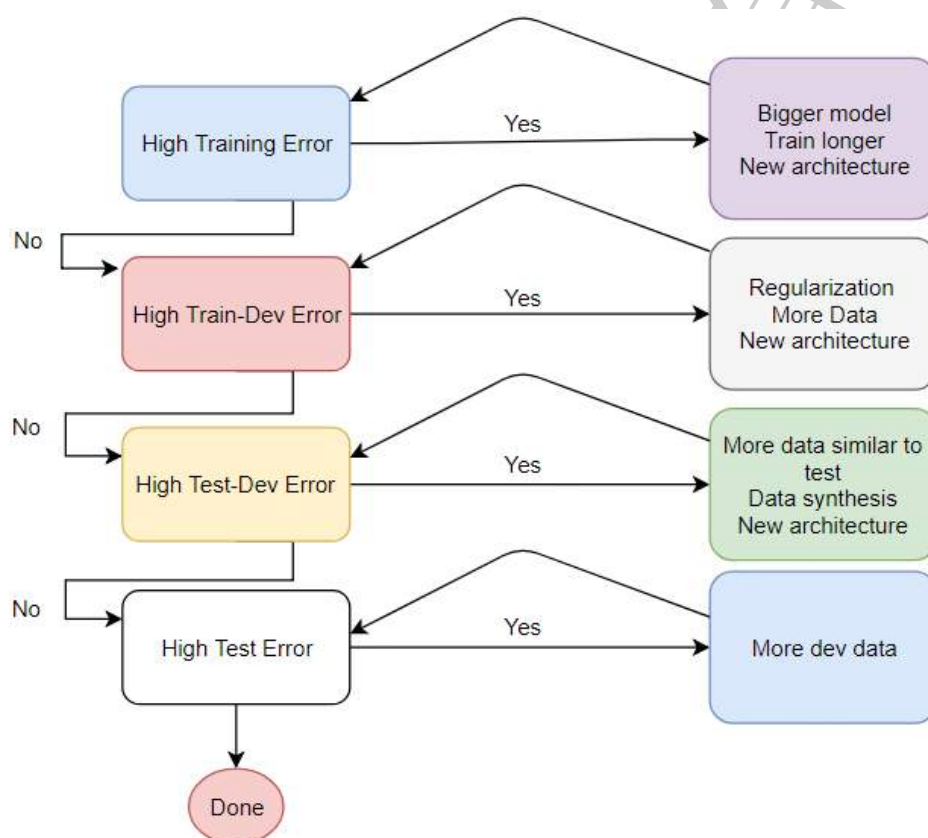


图 25: 深度学习迭代流程

同时与传统的机器学习模型不同，深度学习模型的精度与模型的复杂度、数据量、正则化、数据扩增等因素直接相关。所以当深度学习模型处于不同的阶段（欠拟合、过拟合和完美拟合）的情况下，大家可以知道可以什么角度来继续优化模型。

5.6 本章小节

深度学习作为机器学习的一个分支，具有强大的拟合能力和泛化能力。于此同时深度学习对于数据非常敏感，需要有自己的独特训练方法。针对不同的任务发展有不同类型的深度学习模型，比如图像分类模型、物体检测模型和文本分类模型等，这些模型分别对应于不同的应用方向，有不同的数据

⁴<http://lamda.nju.edu.cn/weixs/project/CNNTricks/CNNTricks.html>

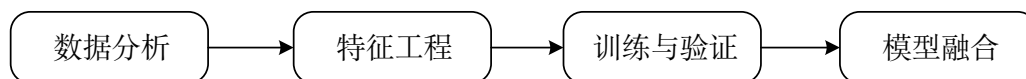
⁵<http://karpathy.github.io/2019/04/25/recipe/>

处理方式和训练技巧。学习深度学习最有效的方法就是结合具体的任务进行学习，在之后的章节中也将深度学习方法与视觉任务和 NLP 任务一同进行讲解。

阿水@版权所有

6 数据挖掘的工作流程

本章的主要内容是解决数据挖掘的流程：数据分析主要目的是分析数据原有的分布和内容；特征工程目的是从数据中抽取出有效的特征；模型训练与验证部分包括数据划分的方法以及数据训练的方法；模型融合部分会简介模型融合的方法和实现方式。



虽然数据挖掘在流程上可以看成是瀑布式的，但各个流程相互影响。比如数据分析可以挖掘出数据的分布规律，可以指导特征工程；特征的验证又可以反馈数据分析的结果。

6.1 数据分析

在拿到数据之后，首先要做的就是数据分析（Exploratory Data Analysis, EDA）。数据分析是数据挖掘中重要的步骤，同时也在其他阶段反复进行。数据就是财富，因为数据本身存在价值，而如何将数据内在的价值充分的挖掘出来，则需要进行有效的数据分析。可以说数据分析是数据挖掘中至关重要的一步，它给之后的步骤提供了改进的方向，也是直接可以理解数据的方式。

在拿到数据之后，我们必须明确以下几件事情：

1. 数据是如何产生的，数据又是如何存储的；
2. 数据是原始数据，还是经过人工处理（二次加工的）；
3. 数据由那些业务背景组成的，数据字段又有什么含义；
4. 数据字段是什么类型的，每个字段的分布是怎样的；
5. 训练集和测试集的数据分布是否有差异；

以上也只是数据分析的冰山一角，数据分析也一直是可以持续的。在数据竞赛中，出题方一般对赛题数据有一些介绍。在数据分析阶段，也需要结合赛题介绍的信息来进行分析。所以你要化身为一个侦探，要有灵敏的嗅觉，关注数据的方方面面。

在分析数据的过程中，还必须要清除的以下数据相关的问题：

1. 数据量是否充分，是否有外部数据可以进行补充；
2. 数据本身是否有噪音，是否需要进行数据清洗和降维操作；
3. 评价函数是什么，和数据字段有什么关系；
4. 数据字段与的标签的关系；

数据分析可以是一个行业、一个岗位，也可以是一个领域。数据分析的形式比较多，且能够用的工具也比较丰富。但数据分析最终目的是想要得到分析得到一些规律，来指导下一步的操作。接下来我将从几个角度来进行具体的讲解，希望能够扩大家对数据分析的视角。

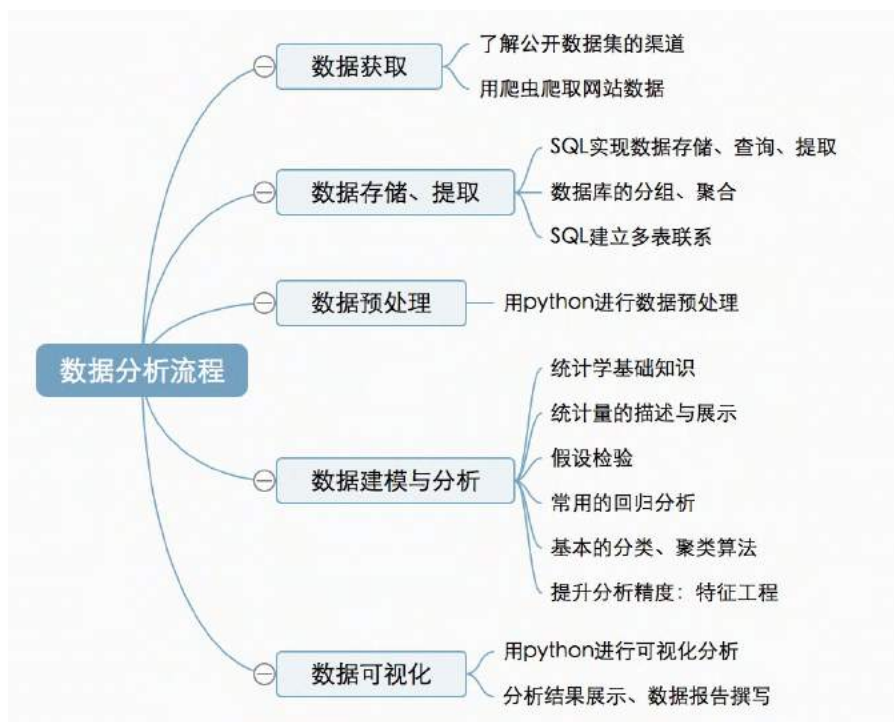


图 26: 数据分析的内容

6.1.1 赛题背景分析

在拿到赛题的数据之后, 首先我们需要对赛题的背景进行理解。这里以 Quick, Draw! Doodle Recognition Challenge 赛题⁶为例, 在拿到赛题后首先需要对赛题的背景和任务进行了解。Doodle Recognition 赛题来自于谷歌之前举办的一个涂鸦游戏⁷, 用户需要绘制给定类别的涂鸦。

在 Doodle Recognition 比赛中, 参赛者需要完成的任务是完成一个模型, 对这些涂鸦结果进行判别分类。赛题使用 MAP@3 进行评价, 也就是预测结果的 Top3 的平均精度。赛题的原始数据是用户绘制的轨迹, 每幅图像可能包括多个轨迹。

在理解 Doodle Recognition 赛题的背景和数据格式后, 可以初步有以下这些结论: 赛题是一个分类问题; 赛题的数据不是一个结构化数据。基于以上对赛题的理解, 可以将赛题抽象成为一个图像分类问题或者序列分类问题, 具体可以用 CNN 或者 RNN 模型解决。

赛题背景分析能够帮助我们理解赛题的任务, 以及赛题数据的收集和评价方法。当然有些赛题的业务逻辑比较简单, 容易理解; 但有一些赛题的业务逻辑经过匿名处理, 就会导致很难对赛题进行理解。无论出题方给定了多少赛题介绍, 参赛选手还是要自己重新理解一遍赛题, 这样可以更加假设赛题的印象。有很多时候, 赛题的一些细节会直接影响到最后的精度, 而这些关键的细节是需要人工发现的。

⁶<https://www.kaggle.com/c/quickdraw-doodle-recognition>

⁷<https://quickdraw.withgoogle.com/>

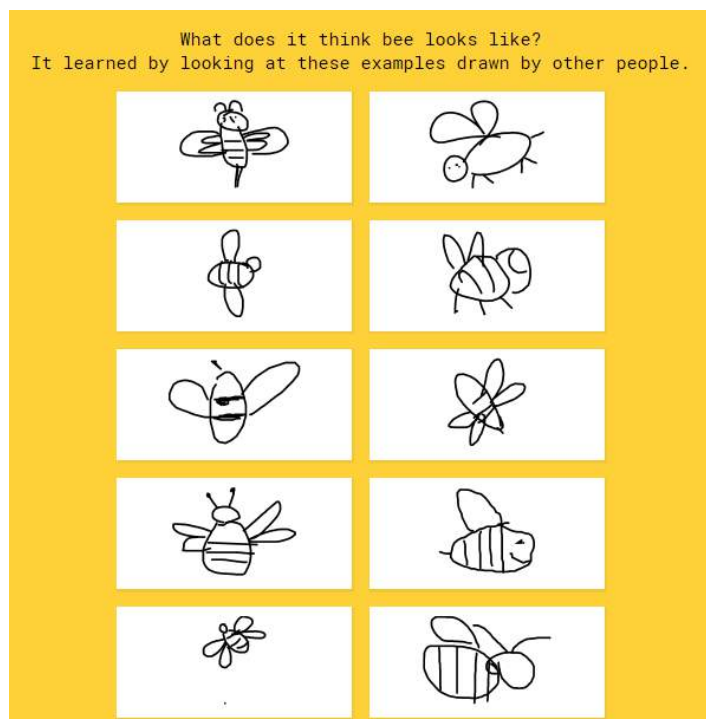


图 27: Quick, Draw! 游戏

6.1.2 赛题数据分析

在有赛题背景知识的基础上，我们就可以进一步对赛题的数据进行了。对于结构化赛题，建模过程主要依赖于人工特征，所以需要对手题数据进行深刻的理解；对于非结构化赛题，建模过程对人工依赖较少，可以利用深度学习强大的建模能力来代替人工特征工程。比如在之前提到的 Doodle Recognition 赛题，我们就可以偷懒，直接使用深度学习完成就够了。

对于结构化数据，我们需要从以下几个角度完成赛题的数据分析：

- 分析数据中每个字段的含义、分布、缺失情况；
- 分析数据中每个字段的与赛题标签的关系；
- 分析数据字段两两之间，或者三者之间的关系；
- 从其他维度分析数据；

因为数据分析本身就是“假设” - “分析” - “验证”的过程，需要分析的目标应该合理，分析的过程需要仔细保存，最后分析的有理可依。同时针对不同类型的数据有不一样的分析方法，有不一样的绘图方法，都是需要注意的地方。

1. 分析数据中每个字段的含义、分布、缺失情况；

- (a) 首先需要结合赛题背景，分析字段的含义，字段表示什么含义、字段的类型是什么、字段的取值空间是什么、字段每个取值表示什么意义。
- (b) 接下来需要分析字段的分布情况，分析字段整体的分布，分析字段在训练集/测试集中的分布情况。

- (c) 最后可以对数据字段进行缺失值得分析，分析字段缺失值的分布比例，字段在训练集/测试集的缺失情况。

2. 分析数据中每个字段的与赛题标签的关系

- (a) 首先可以计算数据每个字段与标签的相关性，可利用相关性计算；
- (b) 接下来可以将字段与标签联合起来进行分析，分析每个字段取值情况下标签的取值情况，分析字段分布于标签的分布情况。

3. 分析数据字段两两之间，或者三者之间的关系

这一步骤与上个步骤类似，也是分析字段与字段之间的关系；

必须要再次强调一点的是，数据分析包括了非常多的细节，可以从很多的角度、很多方式来对数据进行分析。在分析的方式中，绘图最直观最方便的方式。针对不同的数据字段，也有相应的分析方法⁸。

6.2 特征工程

在对数据有一定了解之后，就可以进行下一步的特征工程 (Feature Engineering) 的操作了。特征工程师数据挖掘的核心部分，也是非常依赖人工参数的过程。此外特征工程与具体的机器学习模型联系紧密，不同的机器学习模型对特征有不同的偏好，因此在特征工程阶段需要将字段转成为最合适输入给模型的情况。

特征工程具体包括以下步骤：

1. 数据清洗 (Data Cleaning)

数据清洗主要的目的是提取原始数据中的噪音部分。

2. 特征预处理 (Feature Preprocess)

特征预处理的目的是将数据的原始字段进行相应的编码、变换，并进行缺失值的处理；

3. 特征提取 (Feature Extraction)

特征提取的目的是从原始数据中提取出心的特征字段，并将特征转换成特定的格式；

4. 特征筛选 (Feature Selection)

特征筛选的目的是筛选出较优的特征子集，以取得较好的泛化性能；

6.2.1 数据清洗

数据清洗步骤主要是对数据的噪音进行有效剔除。数据噪音可能有多个来源，来源于数据本身，来源于数据存储，来源于数据转换的过程中。因为噪音会影响特征，也会影响最后的模型结果，因此对数据是非常有必要的。

数据清洗可以从以下几个角度完成：

⁸<https://extremepresentation.typepad.com/blog/2015/01/announcing-the-slide-chooser.html>

1. 对于类别变量，可以统计比较少的取值；
2. 对于数字变量，可以统计特征的分布异常值；
3. 统计字段的缺失比例；

6.2.2 特征预处理

特征预处理包括如下内容：

1. 量纲归一化

(a) 标准化

$$x' = \frac{x - \sigma}{\mu}$$

(b) 区间放缩

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

2. 特征编码

对于**类别特征**来说，有如下处理方式：

- (a) 自然数编码 (Label Encoding)
- (b) 独热编码 (Onehot Encoding)
- (c) 哈希编码 (Hash Encoding)
- (d) 统计编码 (Count Encoding)
- (e) 目标编码 (Target Encoding)
- (f) 嵌入编码 (Embedding Encoding)
- (g) 缺失值编码 (NaN Encoding)
- (h) 多项式编码 (Polynomial Encoding)
- (i) 布尔编码 (Bool Encoding)

对于**数值特征**来说，有如下处理方式：

- (a) 取整 (Rounding)
- (b) 分箱 (Binning)
- (c) 放缩 (Scaling)

3. 缺失值处理

- (a) 用属性所有取值的平均值代替
- (b) 用属性所有取值的中位数代替
- (c) 用属性所有出现次数最多的值代替
- (d) 丢弃属性缺失的样本
- (e) 让模型处理缺失值

6.2.3 特征提取

对于结构化数据来说，在经过有效的特征预处理之后，使用现有的机器学习模型就能够达到比较好的精度了。但是特征预处理步骤只是将特征进行了有效的编码，并没有提取新的特征。还是从业务逻辑上提取一些新的特征，继续挖掘。

在这个步骤没有特定规则，需要具体问题具体分析，还要结合数据分析来完成。总体来提取的新特征应该满足如下要求：

1. 新特征应该是有效的，有业务背景的意义；
2. 新特征不应该加入很多噪音；

6.2.4 特征筛选

在提取完特征后，应该选择有效的特征进行训练模型。为什么不用所有的特征直接进行训练呢，而是选择一部分特征子集来进行训练。原因有如下两点：首先特征本身可能存在噪音，又可能由于分布的原因会影响模型的得分；在原始特征空间中，可能存在冗余特征，会影响模型的训练速度；

特征选择的本质是搜索特征子集，即从原始特征空间中选择出较优的特征集合出来。但特征选择本身就是非常困难的问题，假设原始特征空间的大小为 N ，则可以选择的特征子集个数有 2^N 个；其次特征子集是否有效很难进行评价。

现有的特征选择方式可分为三类：

1. 过滤式选择 (Filter)：特征选择的过程和模型训练过程分开，先进行特征选择，再训练模型；一般使用相关统计量来进行筛选特征⁹：
 - 特征缺失比例；
 - 特征空间方差；
 - 卡方 (Chi2) 检验；
 - Pearson 相关系数；
 - 互信息指数；
2. 包裹式选择 (Wrapper)：将特征选择的过程和模型的训练过程组合，并以模型的性能作为选择特征的依据；所以包裹式可以根据具体的模型进行筛选特征，但需要更多计算量；
3. 嵌入式选择 (Embedded)：先训练好模型，并利用模型参数完成特征选择，筛选过程与过滤式类型，不过嵌入式筛选的标准来自模型的权重系数；

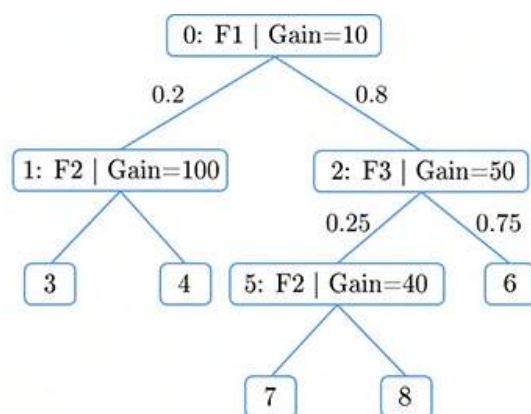
在做数据竞赛的过程中，过滤式选择一般用在初期进行特征筛选，比如从原始数据集中筛选合适的特征子集；包裹式选择一般和模型的训练一起进行，通过模型的性能来筛选特征；嵌入式选择一般用于模型训练后进行特征筛选。需要注意的是特性筛选具有一定的随机性，搜索空间非常大，所以并

⁹https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_selection

没有一个完美的解决方案。在做比赛过程中，验证特征是否有效的方式是包裹式选择，因为模型精度最终的目标，同时包裹式选择的流程可以根据模型来筛选特征，这也是我们需要的。

Trust your local Cross-Validation(CV)，在任何适合都应该优先相信模型交叉验证的精度。

数据竞赛 80% 的时间都花费在包裹式特征上，包裹式选择也是模型训练和验证的过程。特征是否有效、模型是否有效和特征工程是否有效最直接的方式就是精度是否提升。同时还可以借助嵌入式的方法对特征的重要性进行筛选，一般可以采用树模型来完成。树模型拥有强大的建模能力，同时模型的决策过程直观解释性强，所以在 XBoost 和 LightGBM 库中都只带了查看特征重要性的函数。



$\text{FScore}_{F2} = 2$ (appears 2x)

$\text{Gain}_{F2} = \text{Gain}_1 + \text{Gain}_5 = 100 + 40 = 140$

$\text{wFScore}_{F2} = 0.2 + (0.8 \cdot 0.25) = 0.4$ (p to reach node 1 + p to reach node 5)

$\text{ExpectedGain}_{F2} = 0.2 \cdot \text{Gain}_1 + (0.8 \cdot 0.25 \cdot \text{Gain}_5) = 28$

图 28: 树模型信息增益

如果一个特征是有效的，那么树模型会利用该特征完成分裂操作，所以可以计算得到特征的信息增益、特征被分裂的次数、特征被分裂节点的深度和特征是否为叶子节点等信息¹⁰。同时 eli5¹¹和 SHAP¹²做特征重要性可视化也非常方便。

需要注意的是模型权重信息也不一定是合理的。模型的权重是学习出来的，特征的信息增益只能反映特征信息增益的大小，但特征是否为无效特征无从得知。最好的例子就是 Null Importances¹³，随机数也能有很高的信息增益。同时 Null Importances 也是不错的筛选特征方法，值得学习。

6.3 训练与验证

在以上步骤中，我们对数据进行了分析、筛选并根据业务背景提取了新的特征，万里长征终于来到了模型训练与验证的步骤。首先模型的训练与验证并不是一次就够的，是需要反复迭代验证的。在之前的数据清洗特征筛选、特征编码和缺失值编码等众多的步骤中，都是需要验证效果的，那么如何

¹⁰<https://github.com/limexp/xgbfir>

¹¹<https://eli5.readthedocs.io/en/latest/overview.html>

¹²<https://github.com/slundberg/shap>

¹³<https://www.kaggle.com/ogrellier/feature-selection-with-null-importances>

验证这些效果是否有效呢？最直观的方法就是通过验证模型的精度来完成。

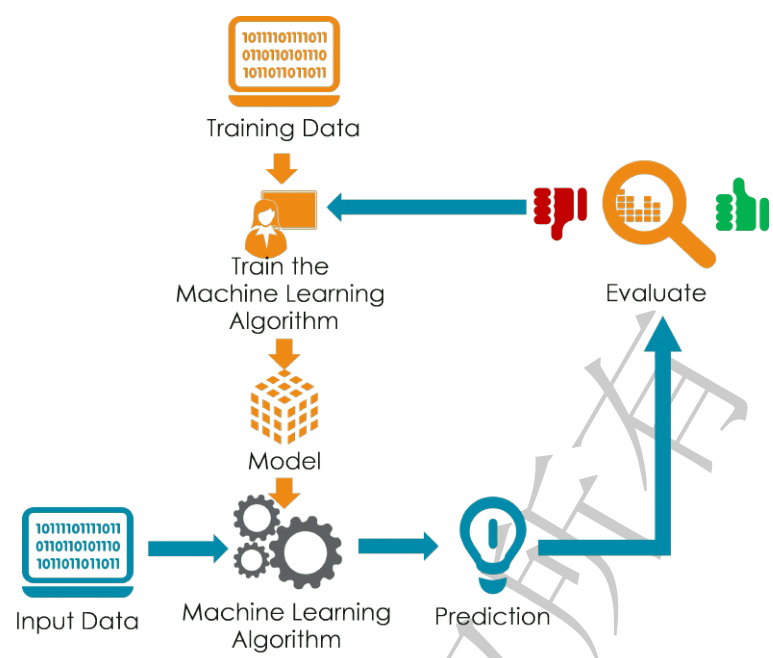


图 29: 机器学习训练评价流程

在给定赛题后，赛题方会给定训练集和测试集两部分数据。参赛者需要在训练集上面构建模型，并在测试集上面验证模型的泛化能力。因此参赛者可以通过提交模型对测试集的预测结果，来验证自己模型的泛化能力。同时参赛方也会限制一些提交的次数限制，以此避免参赛选手“刷分”。

在一般情况下，参赛选手也可以自己在本地划分出一个验证集出来，进行本地验证。训练集、验证集和测试集分别有不同的作用：

- 1. 训练集 (Train Set)：模型用于训练和调整模型参数；
- 2. 验证集 (Validation Set)：用来验证模型精度和调整模型超参数；
- 3. 测试集 (Test Set)：验证模型的泛化能力；

因为训练集和验证集是分开的，所以模型在验证集上面的精度在一定程度上可以反映模型的泛化能力。在划分验证集的时候，需要注意验证集的分布应该与测试集尽量保持一致，不然模型在验证集上的精度就失去了指导意义。

既然验证集这么重要，那么如何划分本地验证集呢。在一些比赛中，赛题方会给定验证集；如果赛题方没有给定验证集，那么参赛选手就需要从训练集中拆分一部分得到验证集。验证集的划分有如下几种方式：

1. 留出法 (Hold-Out)

直接将训练集划分成两部分，新的训练集和验证集。这种划分方式的优点是最为直接简单；缺点是只得到了一份验证集，有可能导致模型在验证集上过拟合。留出法应用场景是数据量比较大的情况。

2. 交叉验证法 (Cross Validation, CV)

将训练集划分成 K 份，将其中的 $K-1$ 份作为训练集，剩余的 1 份作为验证集，循环 K 训练。这种划分方式是所有的训练集都是验证集，最终模型验证精度是 K 份平均得到。这种方式的优点是验证集精度比较可靠，训练 K 次可以得到 K 个有多多样性差异的模型；CV 验证的缺点是需要训练 K 次，不适合数据量很大的情况。

3. 自助采样法 (BootStrap)

有放回的采样方式得到新的训练集和验证集，每次的训练集和验证集都是有区别的。这种划分方式一般适用于数据量较小的情况。

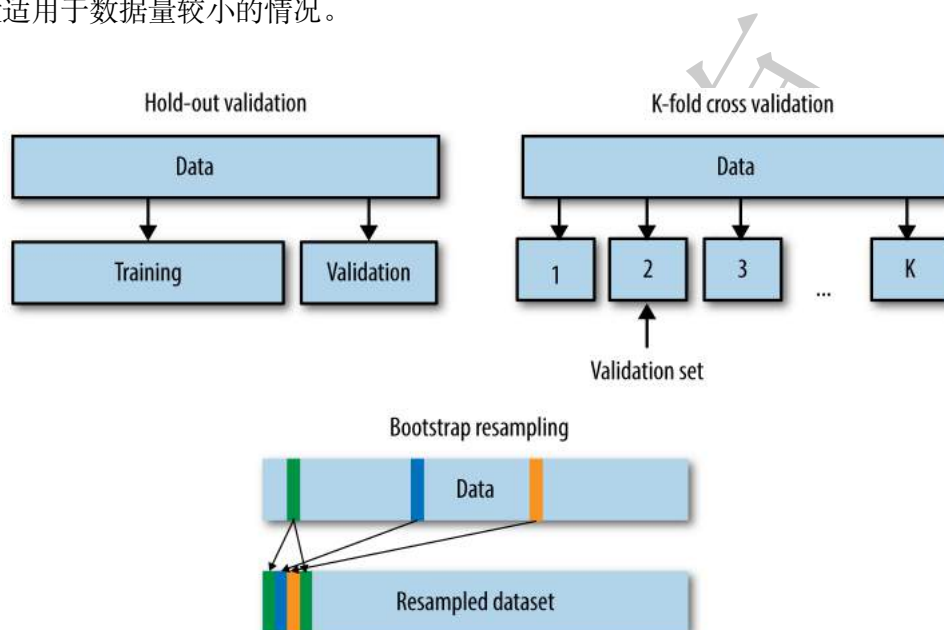


图 30: 数据划分方式

当然这些划分方法是从数据划分方式的角度来讲的，在现有的数据比赛中一般采用的划分方法是留出法和交叉验证法。如果数据量比较大，留出法还是比较合适的。当然任何的验证集的划分得到的验证集都是要保证训练集-验证集-测试集的分布是一致的，所以如果不管划分何种的划分方式，都是需要注意的。这种分布一般指的是与标签相关的统计分布，比如在分类任务中“分布”指的是标签的类别分布，训练集-验证集-测试集的类别分布情况应该大体一致；如果标签是带有时序信息，则验证集和测试集的时间间隔应该保持一致。

此外还需要额外注意数据集是否是带有时序信息，如果有带有时序信息则数据划分还需要考虑时序信息的划分。此外还需要注意标签分布的比例，标签分布可能存在类别不均衡的情况，不均衡的情况可能会影响模型的精度。

Sklearn 中数据划分的函数

在 Sklearn 的 `model_selection` 模块中^a已经定义好常用的数据划分方式，比如交叉验证法、留出法、有放回的采样方法等，使用起来非常方便。

- `train_test_split`: 留出法划分;
- `KFold`: K 折交叉验证;
- `StratifiedKFold`: 分层 K 折交叉验证;

`KFold` 与 `StratifiedKFold` 非常类似，只不过后者是按照标签分类进行划分的，所以每折的样本比例会保持一致。所以当你遇到的数据集的类别分布不太均匀时，建议你采用 `StratifiedKFold`，这样划分的每折数据更加均衡。

^ahttps://scikit-learn.org/stable/modules/classes.html#module-sklearn.model_selection

我们知道常用的机器学习算法的精度和具体的训练数据和超参数相关，而且模型的参数初始化和训练有一定随机性，因此我们如何保证模型可以完美复现呢？这个问题数据竞赛中很重要，为保证模型完美复现，必须要保证数据的划分方式、模型的初始化、模型的超参数都保持一致。

6.4 模型融合

在训练得到模型之后，还可以考虑使用模型融合来继续提高精度，特别是在机器学习竞赛中。原始模型能够决定 90% 的精度，通过模型融合或许能够继续提高 10% 的精度¹⁴。模型融合是基于模型多样性的基础之上的，模型越多样最终融合的结果就越稳定。

可以从偏差与方差的角度来进一步介绍，不同的模型会有不同的偏差和方差，多个模型的融合会减少偏差，最终得到的精度更高的结果。必须要指出的是，模型融合是要求单模型有多样性，铜陵市也要求各个单模型精度都足够好，最终融合模型的结果才会有提高。

模型融合的方式主要有两种：

1. Blending

此种融合方式是直接对模型的输出结果进行融合，是直接对结果文件进行操作的过程。对于分类任务，可以使用多个模型进行投票 (`Vote`) 操作进行融合；对于回归任务，可以使用多个模型结果进行平均 (`Average`) 操作进行融合；对于排序任务，可以使用多个模型的结果进行排序 (`Rank`) 操作进行融合。

`Blending` 操作简单，最终的本地验证结果也更加可靠。

2. Stacking

`Stacking` 操作则是有学习的过程，可以理解为模型结果的二次学习。在验证集划分方式中，我们介绍道 `K-Fold` 交叉划分训练集的方式。通过 `K-Fold` 的划分方式，模型会对训练集的每个部分进

¹⁴<https://mlwave.com/kaggle-ensembling-guide/>

进行一次“独立”的预测 (Out-Of-Fold, OOF)，会对测试集进行 K 次预测。注意这个独立含义，这里的意思是模型把训练集的每一部分都当做未知样本预测了一遍，

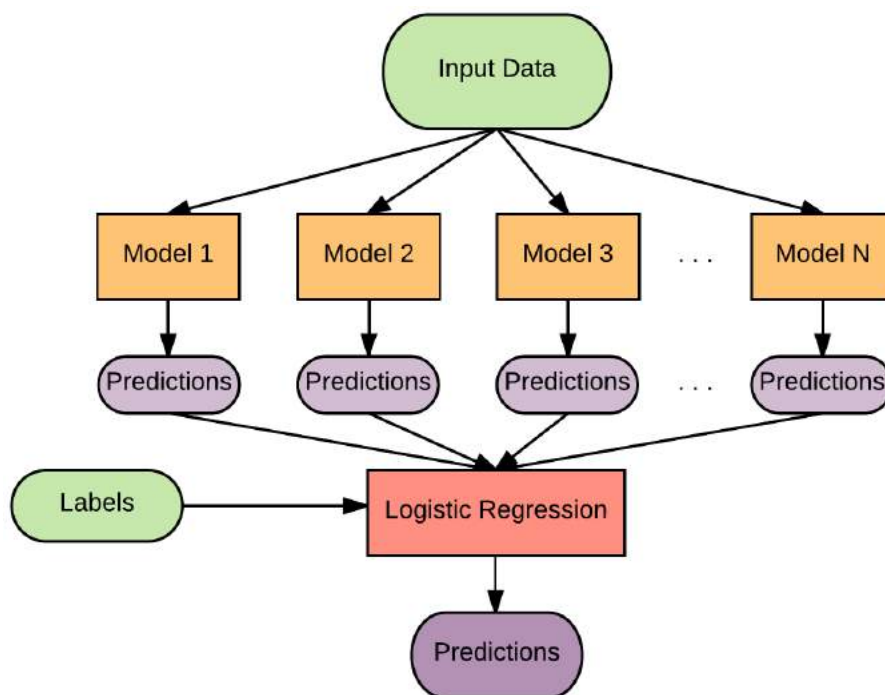


图 31: Stacking 操作

通过 K -Fold 的训练方式，模型会对训练集有一个预测结果，同时对测试集有一个预测结果（将 K 个预测结果进行平均得到）。那么就可以把模型的预测结果看成为一维新的特征，新特征可以加到原始特征进行一起训练。Stacking 将多个模型的新特征拼接一起进行训练，得到最终的预测结果。

在 Stacking 中，模型进行维度可以分为多层，每层的模型不做具体限制。Stacking 本质是 Boosting 的学习思路，利用模型来继续二次学习。Stacking 拟合能力强，但也更容易过拟合。

6.5 本章小结

本章对数据挖掘的流程进行了仔细的讲解，特别是从赛题分析的角度给大家将讲解了数据分析和特征工程的过程。这些过程也是赛题中必要的步骤，也是解决问题的通用步骤，希望大家能够融会贯通。当然最有效的方式还是进行具体的实践，直接拿一个赛题直接开始做。

不同类型的赛题有不同的特点，这些需要积累也需要反复思考。适用于 CTR 类型问题的模型和视觉问题的模型是不一样的，针对具体的问题需要具体分析。

7 结构化数据挖掘

本章的内容是数据结构化数据的解决方法。结构化数据是最常见的数据格式，本质是表格形式的数据。在结构化数据中有很多种应用场景，比如风控场景、价格预测场景、广告预测场景和匿名数据挖掘场景。在本章会以集中比较典型的任务类型为例，以具体的任务给大家进行实战演练。

7.1 常规类型比赛

7.1.1 Two Sigma Connect: Rental Listing Inquiries

<https://www.kaggle.com/c/two-sigma-connect-rental-listing-inquiries>

- **比赛背景**：预测 RentHop 上新的租房信息受欢迎的程度，使用 \logloss 进行评测。
- **比赛数据**：房屋的基本信息（位置、价格、卧室、厕所和文本介绍）、房屋的中介、房屋的照片；
- **解决方案**：提取房屋的“额外”信息，如房屋价格维度、房屋位置维度等维度进行建模；
 1. 房屋的“额外信息”：房屋卧室/厕所的比例、房屋创建的时间、房屋信息完整程度等；
 2. 房屋价格维度：卧室平摊价格、厕所平摊价格、房屋与小区的差价等；
 3. 房屋位置维度：房屋周围地铁站个数、房屋周围公交站个数、房屋周围电影院个数、房屋小区密度；

这道赛题数据量不大，但同时包括了结构化数据、文本数据和图像数据，因此非常适合用于入门学习。同时赛题还包括了 Leak 特征，房屋照片的时间信息与标签强相关，出题方应该是按照时间整理的不同类型的样本。

7.2 CTR 类型比赛

CTR 类型比赛包括点击率预估和广告预测比赛，与转化率相关的场景都可以算作 CTR 类型比赛。同时 CTR 比赛的数据一般是结构化的高维稀疏数据（比如用户 id，商品 id），发展了很多领域模型，如 FM/FFM/FTRL 等。

7.2.1 WSDM-百度好看 APP

Retention Rate of Baidu Hao Kan APP Users

- **比赛背景**：预测使用百度好看 APP 用户第二天是否继续使用 APP，二分类。
- **比赛数据**：用户的基本信息、视频的基本信息和用户在 APP 的行为信息；
- **解决方案**：根据用户的行为习惯提取行为习惯等统计数据，并根据一些 CTR 特征进行建模；

这道赛题的数据量中等，在一台服务器上刚好可以跑，特征工程的步骤可以多线程进行加速操作。同时赛题预测的是用户第二天是否继续使用 APP，所以时间信息就比较关键，赛题还可以做一些 CTR 特征也有不错的效果。

7.2.2 Outbrain Click Prediction

<https://www.kaggle.com/c/outbrain-click-prediction>

- **比赛背景**：预测用户对推荐链接的点击情况，使用 MAP@12 进行评测；
- **比赛数据**：用户的基本信息、网页的信息、广告信息；
- **解决方案**：利用 CTR 思路提取相关特征，使用 FM 进行建模；

7.3 用户信息预测

7.3.1 TalkingData Mobile User Demographics

<https://www.kaggle.com/c/talkingdata-mobile-user-demographics>
TODO

7.3.2 易观-用户性别年龄预测

第二届易观算法大赛自由练习赛——性别年龄预测

- **比赛背景**：预测用户的年龄和性别类别，使用 logloss 进行评测；
- **比赛数据**：用户的手机信息、APP 安装信息、APP 类别信息和用户 APP 使用信息；
- **解决方案**：提取用户的使用习惯和 APP 序列信息，抽象成一个文本分类问题；
赛题本身是一个分类问题，其中 APP 序列就是文本，可以抽象成一个文本分类问题。

7.4 匿名数据挖掘比赛

匿名数据挖掘是指赛题数据的字段是匿名的，也就是数据字段的含义没有进行解释。匿名数据非常常见，由于用户隐私或商业原因，需要对数据进行匿名处理。匿名数据挖掘非常考验数据挖掘和数据分析的能力，非常值得学习。

7.4.1 中诚信征信比赛

中诚信征信比赛是中诚信征信公司于 2017 举办的机器学习比赛

- **比赛背景**：预测用户是否违约，二分类问题，使用 AUC 评测；
- **比赛数据**：匿名用户数据；
- **解决方案**：清洗数据并有效编码数据；

赛题本身是一个典型风控问题，但这个赛题数据包含了很多的噪音和缺失情况，需要仔细进行数据清洗的操作。同时还可以做一些交叉特征。

7.4.2 Allstate Claims Severity

<https://www.kaggle.com/c/allstate-claims-severity>

- **比赛背景：**预测保险索赔单的严重程度，使用 MAE 进行评测。
- **比赛数据：**230 维度的匿名特征，其中类别变量 116 维度，数值变量 14 维度；
- **解决方案：**充分挖掘匿名特征的编码方式，以及交叉特征；

赛题的标签是数值，而且应该是与金额相关的字段，因此可以进行 \log 变换进行转换成类似正态分布的形态，转换后也影响模型的精度。为什么正态化如此重要呢，正态分布是很多模型的先验假设，同时很多模型的参数都是用正态分布初始化的，还有其他的解释¹⁵。

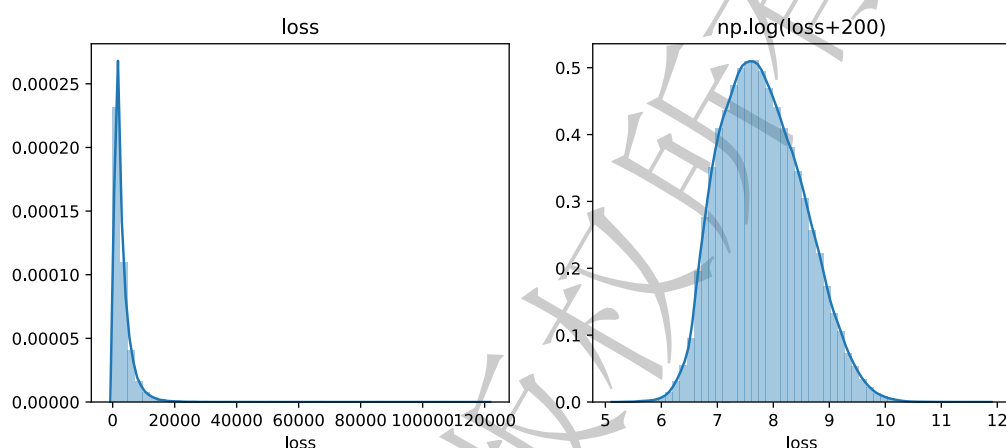


图 32: loss 的变换

此外这道赛题要想取得较好的精度，需要对变量进行有效交叉，以及编码操作。构造的模型至少需要一个树模型和一个神经网络模型进行融合。这道赛题没有 **Leak**，完整复盘能够学习很多特征编码的操作。

7.4.3 Porto Seguro's Safe Driver Prediction

<https://www.kaggle.com/c/porto-seguro-safe-driver-prediction>

- **比赛背景：**预测用户明年是否会提出索赔，分类问题，使用归一化的基尼指数评测。
- **比赛数据：**59 维度的匿名特征，具体包括布尔特征、类别特征和数字特征；
- **解决方案：**将匿名特征进行编码，并训练神经网络模型；

这道赛题数据比较规整，字段类型也比较清晰。基本上简单的特征成果就可以有比较好的结果。比较值得注意的是，赛题的第一名使用降噪自编码器（Denoising Autoencoders, DAE）进行了特征提取，也是非常有学习价值。

¹⁵<https://www.quora.com/Why-does-a-data-scientist-determine-the-distribution-of-data-for-example-with-normality-test-before-starting-to-build-a-machine-learning-model>

7.5 本章小结

在熟悉了上述比赛后相信你会对结构化数据比赛有了新的认识，首先结构化数据比赛在提取特征的过程都是比较类似的，数据的存储方式和提取特征的方式都是大同小异；其次结构化数据比赛可优先选择树模型（LightGBM 或 XGBoost）来完成，也有有一定的套路可寻的。但我们遇到一个新的比赛后，特别是结构化数据挖掘比赛，可以注意以下事项：

1. 对于结构化数据比赛，划分本地验证集至关重要，一定要保证本地验证集和评测方式保持一致。
2. 对于结构化数据比赛，可以优先选取树模型进行建模并进行快速迭代。
3. 对于结构化数据比赛，从特征工程的角度需要注意以下事情：结构化数据比赛的特征工程往往都是类似的，比如都可以提取一些统计特征（最大、最小、平均值、分位数、求和等），这些统计特征都是非常常见的，都可以提取一些与业务相关的特征。

8 计算机视觉任务

本章的内容是计算机视觉的相关任务，从图像处理的基础开始到图像常见的处理方法开始，到图像特征提取算法。计算机视觉任务的本质是如何描述图像，视觉任务可以分为：图像分类、图像检索、人脸检索与识别、物体检测与识别、文本检测与识别等。本章以具体的图像任务（图像分类、图像检索、视频检索）为例给大家讲解常见的解决思路。

计算机视觉的任务就是让机器能够“看”这个世界，更进一步来说计算机视觉就是尝试用计算机来代替人眼，来完成一些特定的任务。计算机视觉的发展以数字图像为基础，因此在本章节会先讲解数字图像的相关基础知识。计算机视觉任务有众多的应用场景，包括工业图像检测、视频事件检测、目标检测和识别、图像检索和三维重建等。本章会以图像特征为侧重点，重点讲解了图像分类、图像检索和视频检索三个部分。

8.1 数字图像处理

数字图像最早起源于 20 世纪初，最早是用于电报行业的图传输。当时图像传播的介质是电缆，也没有完善的图像编码机制，为了保证图像在有限的介质内进行有限传输，人们进而设定了颜色空间和颜色编码的规则。此外其他非可见光也有类似的成像机制，如伽马射线成像、X 射线成像和红外线成像等。

为了将图像进行有效压缩和表示，图像显示都采用量化表示，即用数字来表示原始颜色。在 RGB 色彩空间内，每个颜色空间由 256 个取值组成。图像采用矩阵的形式进行存储，矩阵的大小就是图像的分辨率。所以一幅 1280*768 的彩色图像，就包括 1280*768*3 个像素值（3 表示颜色空间个数）。利用矩阵的形式还有一个优点，就是矩阵运算非常方便。

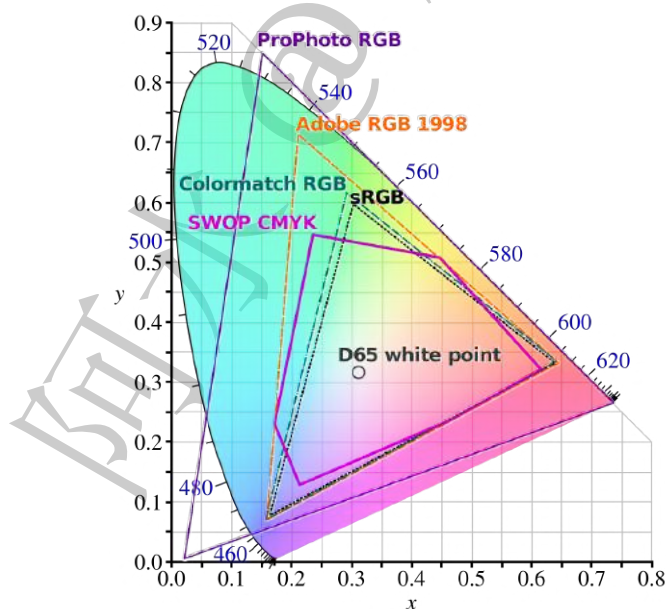


图 33: 颜色空间

1. 颜色空间（Color Space）¹⁶

¹⁶https://en.wikipedia.org/wiki/Color_space

颜色空间是颜色的组织方式，是颜色定义的空间。颜色可以由多种色彩模型来表示，形成各种颜色系统，如 RGB 颜色空间、HSV 颜色空间等。

RGB 颜色空间是将红、绿、蓝三原色的景象组合，以产生各种颜色。RGB 颜色模型被广泛应用于电子信息领域，也是电子显示器的颜色形成方式。

2. 颜色直方图 (Color Histogram) ¹⁷

颜色直方图描述了不同颜色在图片中所占的比例，只统计每个颜色的出现次数。

3. 图像滤波

图像滤波是对图像噪音的处理方式，也可以用于增强图像效果。

8.2 图像特征算子

随着图像应用的越来越广泛，图像数据在生活中随处可见，也给人们带来了一些困扰。比如我用相机连拍了十几张相片，如何剔除差异比较好的相片呢？又比如我有一幅相片，如何找到它的原图呢？类似的问题比比皆是，在数字图像发展的过程中，人们发现只对图像进行存储和编码是不够的，还需要对图像进行有效的表示。

在不同图像任务中，人们定义了不同的图像特征描述方法。但核心的地方都在于如何有效的描述图像，如何编码图像的有效特征。图像可以从全局和布局两个维度进行描述，图像全局特征试图用一些方法来表示图像整体的特征；图像局部特征则关注一些局部区域。

常见的图像全局特征包括：颜色直方图特征、HOG 特征、GIST 特征、LBP 特征等。

1. 颜色直方图特征由于图像是由具体的颜色构成，而不同颜色之间可以用颜色距离来表示相似性。颜色直方图统计的是一幅图像上颜色取值的比例，是颜色空间的分布统计。颜色直方图是一种全局图像特征，不关注具体色彩的空间位置，只对图像整体的颜色空间进行度量，无法描述图像的局部特征。
2. HOG 特征
3. GIST 特征
4. LBP 特征
5. 图像指纹图像指纹定义为图像标识符，用来快速定位和识别图像。图像指纹和文件的 md5sum 值类似，不同的图像应该有不同的指纹。但与 md5sum 不同，图像指纹并不用于校验，而关注图像的内容，即具有相同内容的不同图片文件应该有相似的指纹结果。

图像指纹的计算流程如下：将图片缩放成制定大小的灰度图，然后计算灰度图的颜色均值/频率统计值等，然后将计算结果转换成字符串格式。常见的图像指纹算法包括 dhash、phash 和 whash，具体可以参考 <https://github.com/JohannesBuchner/imagehash> 库。

图像指纹是一个字符串，因此可以直接通过比较字符串来判断两幅图像内容是否相同。此外图像指纹一般由数值转换得来，因此可以使用汉明距离来衡量图像之间的相似性。

¹⁷https://en.wikipedia.org/wiki/Color_histogram

常见的图像局部特征包括：Harris 角点、SIFT 特征、FAST 角点特征等。

1. Harris 角点
2. SIFT 特征

8.3 常见的视觉任务

图像是现实生活中最为常见的多媒体媒介，我们希望计算机能够理解图像以此完成特定的任务。解决视觉任务的流程是提取图像的特征、将特征进行表示，进而将其应用到具体的任务中。图像特征提取和特征表示是两个步骤，前者提取图像的感兴趣的信息点，后者对图像特征进行编码表示。现如今深度学习是一种端到端的过程，网络可以自动完成图像特征提取、编码和映射到具体任务的步骤。

深度学习刷新了很多任务的最高精度，也是现如今最常见最精准的方法了。特别是在非结构化数据（视觉数据和文本数据）上，深度学习具有很高的精度和泛化能力，所以在视觉任务和文本任务上掌握相应的深度学习技巧是必不可少的。

常见的视觉任务包括：

1. 图像分类 (Image Classification)：
2. 图像检索 (Image)：
3. 物体检测 (Object Detection)：
4. 物体分割 (Object Segmentation)：
5. 人体关键点识别 (Pose Estimation)：
6. 字符识别 OCR (Optical Character Recognition)：

8.3.1 图像分类

图像分类是根据图像的语义信息对不同类别图像进行区分，是计算机视觉中重要的基础问题，是物体检测、图像分割、物体跟踪、行为分析、人脸识别等其他高层视觉任务的基础，在许多领域都有着广泛的应用。如：安防领域的人脸识别和智能视频分析等，交通领域的交通场景识别，互联网领域基于内容的图像检索和相册自动归类，医学领域的图像识别等。

在深度学习时代之前，图像分类的解决流程是：提取图像特征、编码特征特征和分类器的训练。在深度学习时代，这些操作都可由神经网络完成。对于图像分类任务，比较经典的数据集是 MNIST 和 CIFAR-10。MNIST 是 28×28 像素的灰度图，具体图像内容是 10 个手写数字；CIFAT 是 32×32 尺寸的彩色图，具体图像内容是常见的 10 种动物。

MNIST 和 CIFAR-10 都是尺寸较小的数据集，没有实际的应用场景。ImageNet 数据集收集了常见的物体 22000 类，数据集图片数量百万级别，数据集图像也非常清晰。在深度学习时代之前，ImageNet 数据集的误差还很高，AlexNet 网络直接将分类误差降低了 10%，这也直接引爆了深度学习浪潮。2011 年至 2017 年，ImageNet LSVRC 比赛每年会举办图像分类、物体定位和视频物体检测等任务，是国内外众多人工智能公司的战场。ImageNet LSVRC 比赛期间出现了众多优秀的图像分类模型，如 VGG、ResNet


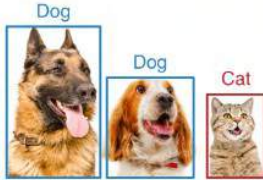

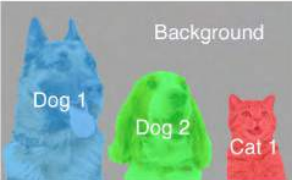

Application	Illustration	Available Models
<p><u>Image Classification:</u> recognize an object in an image.</p>		50+ models, including ResNet , MobileNet , DenseNet , VGG , ...
<p><u>Object Detection:</u> detect multiple objects with their bounding boxes in an image.</p>		Faster RCNN , SSD , Yolo-v3
<p><u>Semantic Segmentation:</u> associate each pixel of an image with a categorical label.</p>		FCN , PSP , DeepLab v3
<p><u>Instance Segmentation:</u> associate each pixel of an image with an instance label.</p>		Mask RCNN
<p><u>Pose Estimation:</u> detect human pose from images.</p>		Simple Pose

图 34: GluonCV 例子

和 SENet 等，直接推动了图像分类任务和物体检测任务的精度。

因为 ImageNet 数据集包含了常见的物体类别，因此在 ImageNet 数据集上训练好的模型可以轻松应用到其他任务上，这就是迁移学习（Transfer learning）的思路。对于深度学习模型可以通过 Fine tune 的方法快速拟合新数据集。

8.3.2 图像检索

随着图像数量不断增多，如何从图片库中快速、准确的检索到感兴趣的图像，逐渐成为信息检索领域的研究热点。图像检索技术可以应用到人脸检索、视频检索、重图检索和安防领域，具有较大的应用场景。

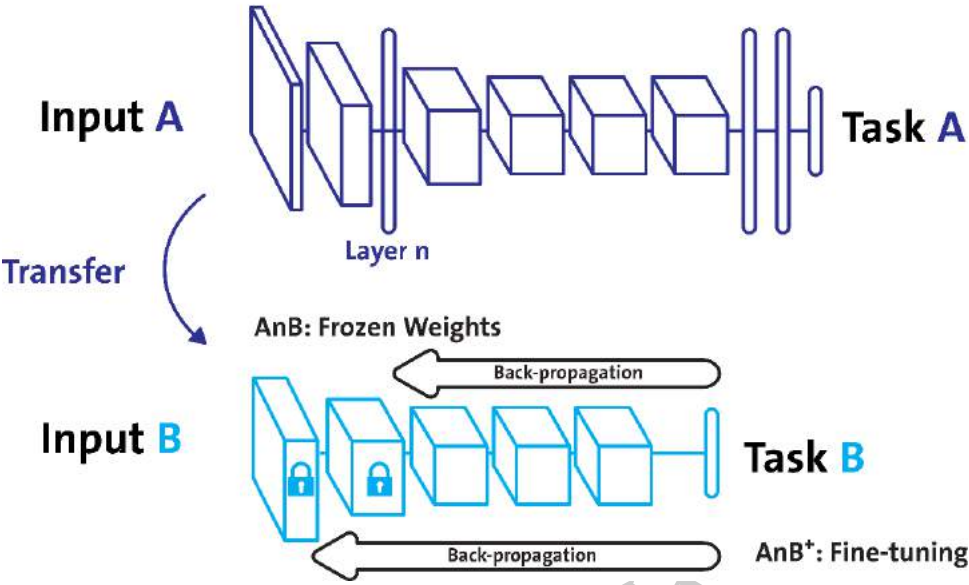


图 35: Fine-tune

图像检索按描述图像内容方式的不同可以分为两类,一类是基于文本的图像检索 (TBIR, Text Based Image Retrieval), 另一类是基于内容的图像检索 (CBIR, Content Based Image Retrieval)。TBIR 起源于上世纪 70 年代,那是互联网网页发展的初期。TBIR 将每个图像打上了文字标签,并通过这些文字标签完成图像检索,将原始问题转换为文字检索的问题。TBIR 思路简单,技术可以直接与搜索引擎技术进行结合,只要对存储图片的文字标签就可以完成图像的检索过程。但 CIBR 也有一些缺点,首先给图片打标签的过程依赖人工,标注的成本比较高;其次不同标注人员对标签的标准可能存在差异,标注人员也不能对图片进行完美的描述。

基于此研究者逐渐尝试使用图像内容来完成图像检索,即 CBIR 技术。CBIR 检索的思路立足于图像的内容,利用图像的内容特征来完成检索。CBIR 首先会利用图像特征提取算法提取图像库中所有图像的特征,并进行有效存储编码;但遇到需要检索需求,先将待检索图像利用相同的特征算法提取并编码图像特征,并将检索特征与原有图片库的特征空间进行比对,找到与待检索图像显示的结果。

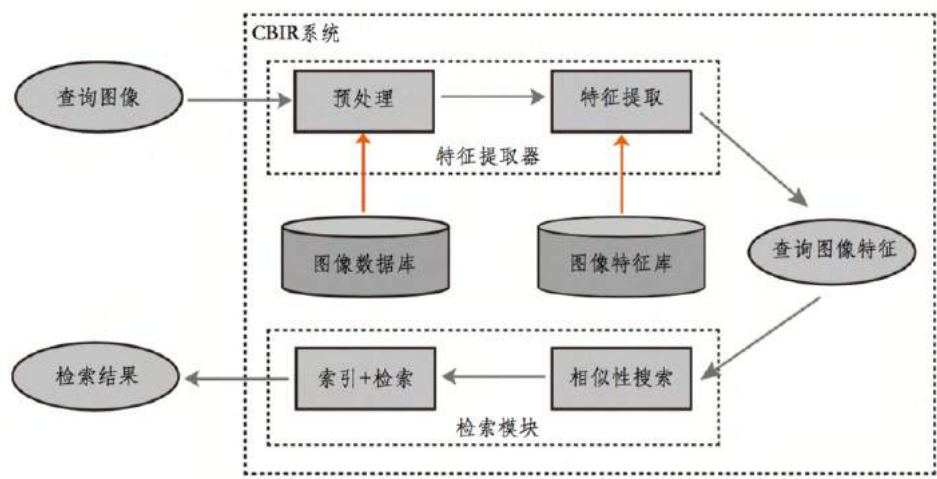


图 36: CBIR 检索流程

与 TBIR 相比, CBIR 有以下优点: CBIR 技术立足于图像的内容, 不局限于图像标签, 因此可以从多个维度(颜色、纹理、物体等)来提取图像特征; CBIR 技术, 不依赖于人工, 方面进行大规模应用。

CBIR 技术有三大关键点:

1. 如何提取图像的关键特征;
2. 如何对提取到的特征进行有效编码;
3. 如何根据特征进行匹配;

在深度学习之前的时代, CBIR 技术大都利用图像局部特征(如 SIFT 特征)来完成图像特征的提取, 并借助特征量化的方式将图像特征映射到特征空间, 并借助于特征距离完成图像特征的检索过程。随着深度学习在图像分类任务中的崛起, 研究者发现卷积神经网络是很有效的图像特征提取方式, 图像检索也逐渐进入了深度学习时代。

根据检索的场景, 可以将图像检索任务分为以下三种任务:

1. 基于类别(Class)的检索任务: 目标找到相同类别的结果;
2. 基于实例(Instances)的检索任务: 目标找到包含相同物体的结果
3. 基于拷贝(Copy)的检索任务: 目标找到相同图像或经过变换的结果;

图像检索有广泛的应用场景, 是计算机视觉类型任务的基础。例如视频检索的本质还是图像特征相似度, 人脸检索任务的核心也是人脸图像相似度。

8.3.3 物体检测

TODO

8.3.4 字符识别

1929 年, 德国科学家 Tausheck 提出了 OCR 的概念, 定义为将印刷体的字符从纸质文档中识别出来。而技术发展至今, 识字, 已不再仅限于识别书本上的文字, 而是要识别真实世界开放场景中的文字。因此, 也衍生出了一系列问题, 例如真实环境中文字角度不可控、语种复杂多样、环境噪声多变等, 针对这些问题, 学术界开展了 OCR 领域研究工作。

8.4 视觉类型比赛

视觉类型的比赛一般以图像数据出现, 比赛任务包括常见的视觉任务, 如图像分类、地标分类、图像检索、人脸检索、遥感图像分类、图像/视频物体识别和图像超分辨率重建。视觉类型的比赛比较多样, 基本上每一种类型的比赛都对应一个视觉任务方向。此外若数据可以转换成矩阵形式, 那么也一般也将其视为视觉比赛。

视觉类型比较的技巧

对于视觉类型比赛来说一个建模过程(好的 **baseline** 模型)和完成的解题流程非常重要, **baseline** 模型影响了你初始的得分, 解题流程决定了迭代的速度。

- 赛题建模: 你把赛题任务当做何种任务, 分类? 识别? 检索?。对应的使用哪些模型?
- 解题流程: 赛题具体包括哪些步骤, 数据如何预处理? 如何调参?

为什么视觉比赛的建模过程非常重要呢, 现如今不同视觉任务都有常见的 **baseline** 模型, 而不同的 **baseline** 模型直接就会导致精度的差异。其次视觉比赛训练模型周期都非常长, 为了快速迭代应该解题步骤尽量拆分。

一般情况下视觉任务包括: 数据预处理、模型训练和测试集预测三个步骤。针对不同的任务有不同的数据预处理和数据增强技巧, 对应的也有相应的模型调参技巧, 参赛时可以查阅历史类似比赛的经验进行学习。此外测试集增强 (Test Time Augmentation, TTA) 也非常关键, 也会影响最终的精度。

8.4.1 Quick, Draw! Doodle Recognition Challenge

<https://www.kaggle.com/c/quickdraw-doodle-recognition>

- **比赛背景:** 对 Quick, Draw! 任务中的涂鸦进行分类, 使用 MAP@3 进行评测;
- **比赛数据:** 用户绘制的轨迹、是否被识别和用户地区编号;
- **解决方案:** 将用户的绘制轨迹视为一个图像问题或者序列问题, 训练一个深度学习模型进行分类; 赛题任务比较简单是一个多分类问题, 具体包括 340 类。赛题原始数据是用户绘制的轨迹, 是带有空间位置的字符串序列。可以把赛题抽象成一个序列数据的分类问题, 也可以抽象成一个图像分类问题; 针对前种建模方式可使用 RNN/LSTM/GRU 等模型进行解决, 针对后种情况则可以使用 CNN 模型进行解决。

对赛题进行建模后就是实践的细节了, 赛题数据如何读取、模型如何训练和预测结果如何进行增强, 这每一个操作都会影响最终的精度。首先使用深度网络来进行建模, 就必须思考数据如何进行扩增的问题; 其次赛题的数据量比较大、数据尺寸也比较大, 训练深度网络容易不收敛, 因此可以先在小数据集上进行训练, 然后迁移到全量数据集上; 最后赛题是多分类问题, 所以标签的分布也会影响最终的得分, 对于提交结果可以进行类别“平滑”操作。

8.4.2 Google Landmark Retrieval Challenge

TODO

8.4.3 Google Landmark Recognition Challenge

TODO

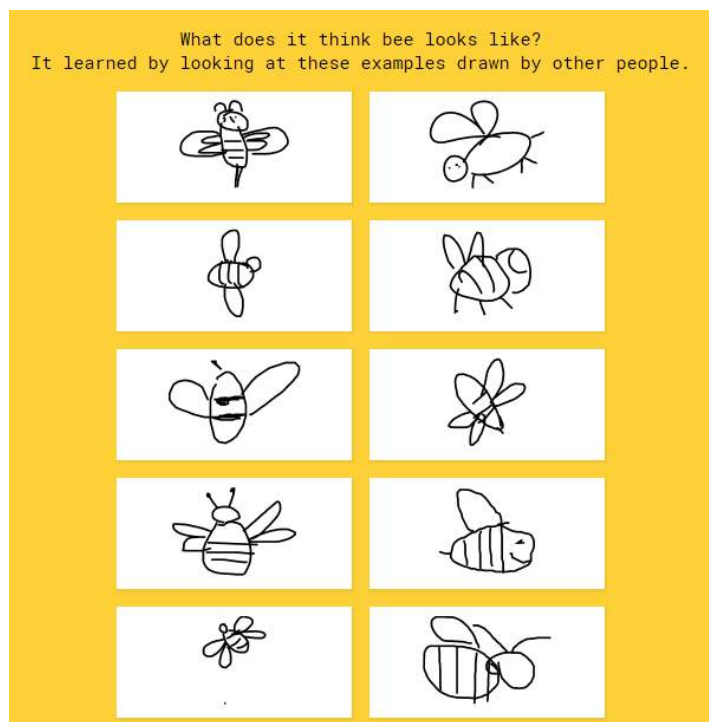


图 37: Quick, Draw! 样例

8.4.4 TinyMind 人民币面值 & 冠字号编码识别挑战赛

<https://www.tinymind.cn/competitions/47>

- **比赛背景：**任务 1 对人民币的类别进行分类，任务 2 并对人民币的编号进行识别；
- **比赛数据：**4W 张人民币图片；
- **解决方案：**任务 1 使用分类模型；任务 2 使用 OCR 模型；



图 38: TinyMind 人民币数据样例 (类别为 2 元，编码为 T184140842)

这个赛题是非常常规的视觉任务，任务 1 是对人民币的类别进行分类，任务 2 是对人民币的编码进行识别。当遇到一个视觉比赛时应该将问题定义好，也就是说将其抽象成哪一种具体的视觉任

务，进而使用对应的视觉模型进行解决。赛题的任务 1 很明显是一个图像的多分类问题，所以使用 CNN 完成分类即可。任务 1 需要对图片进行分类，直接 fine-tune 一个分类网络效果就很好了。任务 2 则稍微复杂一点，需要对人民币的编码进行识别。对于需要识别的任务来说，任务本质还是一个检测 + 识别的过程。赛题的任务 2 也可以说是一个 OCR 问题，也包括检测和识别两个过程。检测识别图像中感兴趣的区域，识别进而对区域内的内容进行识别。在 OCR 问题中通常使用 EAST 模型进行检测，使用 CRNN 模型进行识别。EAST 模型是专门用于检测文本区域的模型，可以检测多尺度多方向的模型。但对于本赛题不需要特别强大的检测模型，因为编码一般都是水平方向的且大小比较固定，所以可以直接使用通用的检测模型 Faster-RCNN 代替。

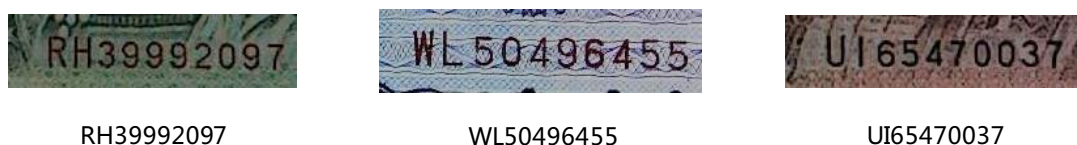


图 39: TinyMind 人民币数据样例 (检测后效果)

当检测过程完成后就进行识别过程，字符识别其实已经是比较成熟的领域了，研究的比较完备。对于任务 2 的识别问题，具体可以把它看做是一个定长字符的识别问题，具体可以使用 CRNN 模型或者 muti-CNN 来完成。其中 muti-CNN 是一个 CNN 完成多个分类的过程，传统 CNN 分类网络可以进行单个字符分类，可以在分类 CNN 的卷积层后面接多个全连接层完成多分类的操作。为什么要多个全连接层呢，这里是把每个字符的识别抽象成一个分类问题，而每个字符的识别视为相互独立的。

muti-CNN 是比较简单的识别定长字符的方法，思路简单训练起来也非常直接，只需要对分类模型进行简单的改进即可。但这种方法也有一些缺点，首先 muti-CNN 抽象成单个字符的识别过程，所以没有考虑字符与字符之间的关系；其次字符识别如果直接用分个多字符进行训练，容易导致模型过拟合。

CRNN 模型由 CNN+RNN 的构成，先利用 CNN 提取图像的特征再通过 RNN 对内容进行识别。CRNN 本质也是一个分类模型，RNN 部分是使用 CNN 特征进行分类操作，分类损失由 CTC 计算，非常适合非定长的字符识别场景。

8.4.5 Urban Region Function Classification

TODO

8.5 本章小节

TODO

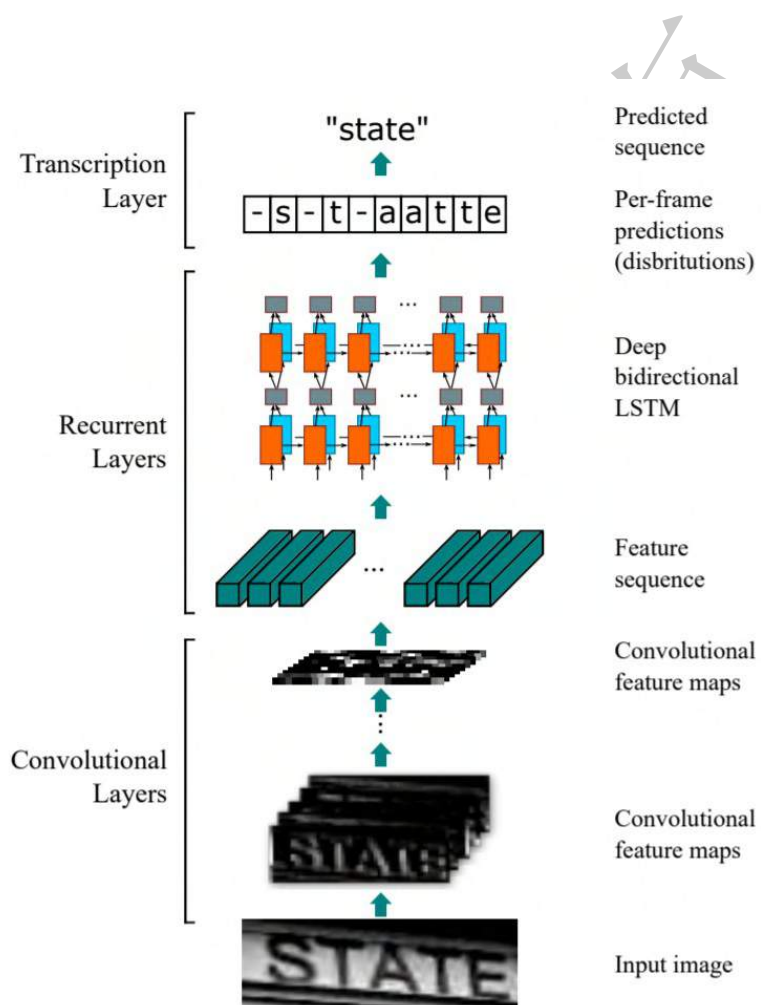


图 40: CRNN 网络结构

9 自然语言处理任务

本章的内容是自然语言处理 (Natural Language Processing, NLP)¹⁸, 主要以 NLP 的相关任务为主讲解 NLP 的相关知识。NLP 是语言学与人工智能的分支, 试图让计算机能够完成处理语言、理解语言和生成语言等任务。由于篇幅原因, 本章具体以中文分词、词性标注、垃圾文本分类和文本情感分类为例。

自然语言处理是研究计算机处理人类语言的一门技术, 具体包括:

1. 词法分析 (Lexical Analysis): 对自然语言进行词汇层面的分析, 是 NLP 基础性工作

- 分词 (Word Segmentation/Tokenization): 对没有明显边界的文本进行切分, 得到词序列
- 新词发现 (New Words Identification): 找出文本中具有新形势、新意义或是新用法的词
- 形态分析 (Morphological Analysis): 分析单词的形态组成, 包括词干 (Stems)、词根 (Roots)、词缀 (Prefixes and Suffixes) 等
- 词性标注 (Part-of-speech Tagging): 确定文本中每个词的词性。词性包括动词 (Verb)、名词 (Noun)、代词 (pronoun) 等
- 拼写校正 (Spelling Correction): 找出拼写错误的词并进行纠正

2. 句子分析 (Sentence Analysis): 对自然语言进行句子层面的分析, 包括句法分析和其他句子级别的分析任务

- 组块分析 (Chunking): 标出句子中的短语块, 例如名词短语 (NP), 动词短语 (VP) 等
- 超级标签标注 (Super Tagging): 给每个句子中的每个词标注上超级标签, 超级标签是句法树中与该词相关的树形结构
- 成分句法分析 (Constituency Parsing): 分析句子的成分, 给出一棵树由终结符和非终结符构成的句法树
- 依存句法分析 (Dependency Parsing): 分析句子中词与词之间的依存关系, 给一棵由词语依存关系构成的依存句法树
- 语言模型 (Language Modeling): 对给定的一个句子进行打分, 该分数代表句子合理性 (流畅度) 的程度
- 语种识别 (Language Identification): 给定一段文本, 确定该文本属于哪个语种
- 句子边界检测 (Sentence Boundary Detection): 给没有明显句子边界的文本加边界

3. 语义分析 (Semantic Analysis): 对给定文本进行分析和理解, 形成能够表达语义的形式化表示或分布式表示

- 词义消歧 (Word Sense Disambiguation): 对有歧义的词, 确定其准确的词义
- 语义角色标注 (Semantic Role Labeling): 标注句子中的语义角色类标, 语义角色, 语义角色包括施事、受事、影响等
- 抽象语义表示分析 (Abstract Meaning Representation Parsing): AMR 是一种抽象语义表示形式, AMR parser 把句子解析成 AMR 结构

¹⁸https://en.wikipedia.org/wiki/Natural_language_processing

- 一阶谓词逻辑演算 (First Order Predicate Calculus): 使用一阶谓词逻辑系统表达语义
- 框架语义分析 (Frame Semantic Parsing): 根据框架语义学的观点, 对句子进行语义分析
- 词汇/句子/段落的向量化表示 (Word/Sentence/Paragraph Vector): 研究词汇、句子、段落的向量化方法, 向量的性质和应用

4. 信息抽取 (Information Extraction): 从无结构文本中抽取结构化的信息

- 命名实体识别 (Named Entity Recognition): 从文本中识别出命名实体, 实体一般包括人名、地名、机构名、时间、日期、货币、百分比等
- 实体消歧 (Entity Disambiguation): 确定实体指代的现实世界中的对象
- 术语抽取 (Terminology/Glossary Extraction): 从文本中确定术语
- 共指消解 (Coreference Resolution): 确定不同实体的等价描述, 包括代词消解和名词消解
- 关系抽取 (Relationship Extraction): 确定文本中两个实体之间的关系类型
- 事件抽取 (Event Extraction): 从无结构的文本中抽取结构化事件
- 情感分析 (Sentiment Analysis): 对文本的主观性情绪进行提取
- 意图识别 (Intent Detection): 对话系统中的一个重要模块, 对用户给定的对话内容进行分析, 识别用户意图
- 槽位填充 (Slot Filling): 对话系统中的一个重要模块, 从对话内容中分析出于用户意图相关的有效信息

5. 顶层任务 (High-level Tasks): 直接面向普通用户, 提供自然语言处理产品服务的系统级任务, 会用到多个层面的自然语言处理技术

- 机器翻译 (Machine Translation): 通过计算机自动化的把一种语言翻译成另外一种语言
- 文本摘要 (Text summarization/Simplification): 对较长文本进行内容梗概的提取
- 问答系统 (Question-Answering System): 针对用户提出的问题, 系统给出相应的答案
- 对话系统 (Dialogue System): 能够与用户进行聊天对话, 从对话中捕获用户的意图, 并分析执行
- 阅读理解 (Reading Comprehension): 机器阅读完一篇文章后, 给定一些文章相关问题, 机器能够回答
- 自动文章分级 (Automatic Essay Grading): 给定一篇文章, 对文章的质量进行打分或分级

上述将 NLP 相关任务按照按照层次进行了划分, 这些任务的目的都非常直观。也可以将 NLP 任务可以分为四类:

1. 序列标注: 比如中文分词, 词性标注, 命名实体识别, 语义角色标注等都可以归入这一类问题。这类任务的共同点是句子中每个单词要求模型根据上下文都要给出一个分类类别;
2. 分类任务: 比如我们常见的文本分类, 情感计算等都可以归入这一类。这类任务特点是不管文章有多长, 总体给出一个分类类别即可;
3. 句子关系判断: 比如问答推理, 语义改写, 自然语言推理等任务都是这个模式, 它的特点是给定两个句子, 模型判断出两个句子是否具备某种语义关系;

4. 生成式任务：比如机器翻译，文本摘要，写诗造句，看图说话等都属于这一类。它的特点是输入文本内容后，需要自主生成另外一段文字。

图像和自然语言作为重要的信息载体，是非常重要的研究方向。与计算机视觉（CV）相比 NLP 的任务更加基础和清晰，NLP 的发展也比较早很多任务和工具都已经比较成熟了。一张图片的每个像素点可能并没有具体的含义，但一段句子中的每个字/词都有具体的含义。一张图片加入一些噪音或者进行灰度变换并不会减少图像的语义内容，但一段句子如果加入了噪音或者剔除一些单词就会直接影响句子的含义。设想一下“我欠你一百万”和“你欠我一百万”两个句子就两个字符换个位置，句子的含义就变的天翻地覆了。

各种 NLP 任务虽然看起来差异很大任务各不相同，但本质还是要理解单词和句子。所以如果我们能够让计算机理解句子，这些任务自然都迎刃而解了。

9.1 文本分词与词性标注

分词是自然语言处理的一个非常重要的组成部分，在英文中使用空格分隔词语，在中文中这需要进行更加复杂的操作。分词的目的是将不同的词分隔开，将句子分解为词和标点符号。中文分词算法主要有两种：词典分词算法和统计分词算法。词典分词算法也称字符串匹配算法，是按照一定的将待匹配的字符串与已建立好的词典中的词进行匹配，若找到某个词条，则说明识别了该词。统计分词算法借助机器学习模型完成训练，并根据单词上下文计算单词出现的概率。词典分词算法思路简单，也没有训练的过程；统计分词算法根据词语上下文分词，有较好的泛化性能。

词性标注（part-of-speech tagging），又称为词类标注或者简称标注，是指为分词结果中的每个单词标注一个正确的词性的程序，也即确定每个词是名词、动词、形容词或者其他词性的过程。词性标注这里基本可以照搬分词的工作，在语言中大多数词语只有一个词性，或者出现频次最高的词性远远高于第二位的词性。据说单纯选取最高频词性，就能实现 80% 准确率的中文词性标注程序。

词性标注可以分为基于规则和基于统计的方法，下面列举几种统计方法：

- 基于最大熵的词性标注
- 基于统计最大概率输出词性
- 基于 HMM 的词性标注

9.2 文本预训练技术

预训练过程是深度学习在图像或者视频领域的一种比较常规方法，能够提高任务精度并减少训练时间。在视觉领域我们设计好网络结构以后，可以先用某个训练集合比如训练集合 A 或者训练集合 B 对这个网络进行预先训练，在 A 任务上或者 B 任务上学会网络参数，然后存起来以备后用。假设我们面临第三个任务 C，网络结构采取相同的网络结构，在比较浅的几层 CNN 结构，网络参数初始化的时候可以加载 A 任务或者 B 任务学习好的参数，其它 CNN 高层参数仍然随机初始化。之后我们用 C 任务的训练数据来训练网络，此时有两种做法，一种是浅层加载的参数在训练 C 任务过程中不动，这种方法被称为“Frozen”；另外一种底层网络参数尽管被初始化了，在 C 任务训练过程中仍然随着训练的进程不断改变，这种一般叫“Fine-Tuning”，顾名思义，就是更好地把参数进行调整使得更适应当前的 C 任务。一般图像或者视频领域要做预训练一般都这么做。

预训练方式有几个优点：首先，如果手头任务 C 的训练集合数据量较少的话，现阶段的好用的 CNN 比如 Resnet/Densenet/Inception 等网络结构层数很深，几百万上千万参数量算起步价，上亿参数的也很常见，训练数据少很难很好地训练这么复杂的网络，但是如果其中大量参数通过大的训练集合比如 ImageNet 预先训练好直接拿来初始化大部分网络结构参数，然后再用 C 任务手头比较可怜的数据量上 Fine-tuning 过程去调整参数让它们更适合解决 C 任务，那事情就好办多了。这样原先训练不了的任务就能解决了，即使手头任务训练数据也不少，加个预训练过程也能极大加快任务训练的收敛速度，所以这种预训练方式是老少皆宜的解决方案，另外疗效又好，所以在做图像处理领域很快就流行开来。

那么预训练为什么起作用呢？主要的原因是预训练好的网络参数，尤其是底层的网络参数抽取出特征是和具体任务越无关的，具备任务的通用性，所以这是为何一般用底层预训练好的参数初始化新任务网络参数的原因。而高层特征跟任务关联较大，实际可以不用使用，或者采用 Fine-tuning 用新数据集清洗掉高层无关的特征抽取器。

9.3 常见的 NLP 任务

9.3.1 文本分类任务

垃圾文本分类是 NLP 中经典的分类，目的是将文本分为垃圾文本和非垃圾文本两类。NLP 的分类任务在思路上都比较类似，在流程和方法上都可以相互借鉴。

文本分类算法可以分为三类：

- 基于人工规则的方法：基于人工关键词和规则的匹配来完成。优点是思路简单清晰；缺点是需要花费很多人力成本，精度有限；
- 传统机器学习方法：基于统计语言模型和 SVM/LR/树模型来完成。
- 深度学习方法：基于词向量和深度学习模型完成。

9.3.2 文本信息抽取

9.4 NLP 类型比赛

9.4.1 第三届阿里云安全算法挑战赛

TODO

9.4.2 第二届易观算法大赛-性别年龄预测

TODO

9.4.3 科大讯飞-大数据应用分类标注挑战赛

TODO

9.5 本章小节

TODO

企业IPO招股说明书关键信息抽取示例：

肖萍、李清文夫妇系公司实际控制人，合计控制公司95.00%的表决权，其中直接持有公司9.36%的股权，通过泰萍鼎盛、奕龙达克、创新一号和创新二号控制公司85.64%的表决权。肖萍先生，出生于1974年9月，中国国籍，无境外永久居留权，身份证号码为36031119740904****。李清文女士，出生于1976年6月，中国国籍，无境外永久居留权，身份证号码为36030219760603****。

*抽取实际控制人直接及间接持股比例，国籍，姓名，身份证号

法院裁判文书案情要素抽取示例：

内蒙古自治区鄂尔多斯市中级人民法院审理鄂尔多斯市人民检察院指控被告人王青志犯抢劫罪一案，于2011年11月9日以（2011）鄂刑二初字第15号刑事附带民事判决，认定被告人王青志犯抢劫罪，判处死刑，剥夺政治权利终身，并处没收个人全部财产。

*抽取公诉方，被告人及判决结果

买卖合同关键信息抽取示例：

合同生效、价款及支付
4.1 本合同限于首钢CCPP 110KV GIS项目，仅在此项目下产生法律效力。
4.2 合同价款：小写：（人民币）¥123,961.36
大写：（人民币）壹拾贰万叁仟玖佰陆拾壹元叁角陆分
4.3 合同价款构成：税费、物资费、运保费、包装费、指导、安装调试费等。
4.4 价款支付方式和时间：货到买方验收合格后60天付100%全款。

*抽取款项币种，合同价款，付款条件，付款方式，付款比例，合同金额包含包装费用

图 41: 文本信息抽取例子

10 其他相关任务

10.1 AutoML

随着机器学习在众多领域内被广泛应用，人们发现机器学习的从业者在学习过程中有着不可替代的作用。AutoML (Automated Machine Learning, 自动机器学习) 目标是降低机器学习的从业门槛，提供能够自动完成机器学习任务的工具，最大程度的减少机器学习专家的介入。

在现有的机器学习/数据挖掘的优化过程中，每个流程都需要有人工的介入：

- 数据分析阶段：需要人工进行绘图分析和数据预处理操作；
- 特征工程阶段：需要人工特征工程和特征筛选；
- 模型训练阶段：需要人工选择合适的机器学习模型和参数；

所以 AutoML 的任务就是尽可能的替代人的操作，减少人工的参与。只要有人工的参与存在，AutoML 就有对应的应用场景。解决一个机器学习任务其实本质都是有一些规律的，AutoML 的目标是减少算法工程师的参与。所以我给大家来分析下完成一个机器学习任务，算法工程师需要做什么：收集整理数据、训练验证模型、模型调参再训练，如此迭代至模型满足要求。一个合格的机器学习算法工程师是知道该在什么阶段做什么操作的，合格的 AutoML 系统也是如此。AutoML 将人工的参与转换成一个空间搜索和迭代的过程，不断的搜索得到最优的解决方案。

AutoML 是一个快速发展的年轻领域，在各个方向上都有对于的应用场景¹⁹²⁰。

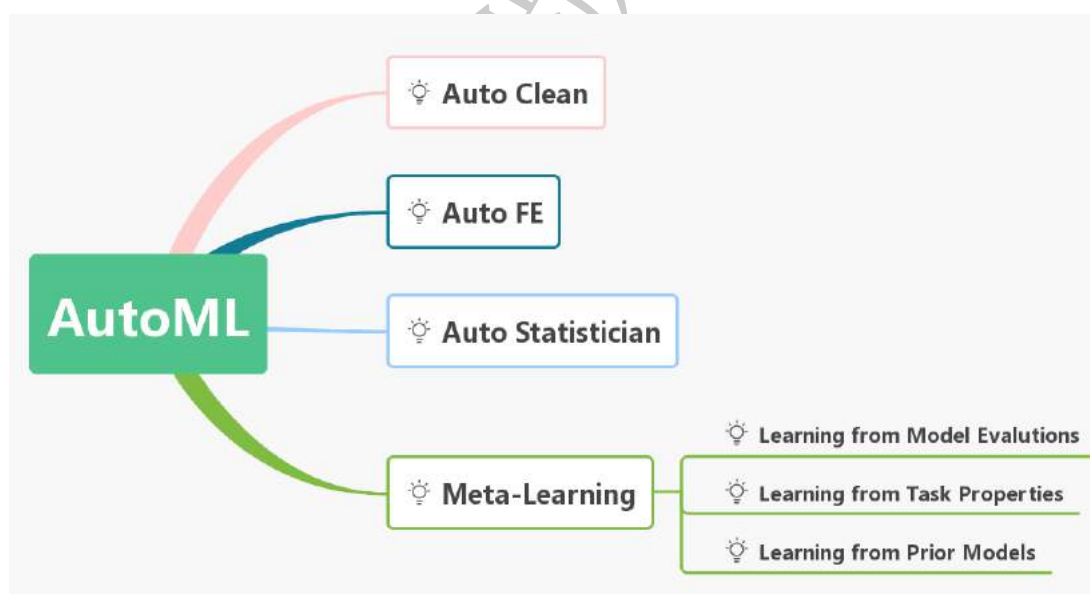


图 42: AutoML 分支

天下没有免费的午餐，也没有完全适用任何场景的机器学习算法。为了在不同数据集上得到较优的结果，必须要针对每个数据集构建流程。这个构建需要人工的才做，就有很高的人力成本和实践成本，AutoML 的目标就是解决此类问题。AutoML 并非是一个新领域，在上个世纪就有模型超参数优化

¹⁹<https://www.automl.org/automl/>

²⁰<https://github.com/hibayesian/awesome-automl-papers>

(Hyperparameter Optimization) 的相关工作，比如贝叶斯调参²¹方法就能够在有限时间内搜索得到比较好的模型参数。

AutoML 还可以应用到工程阶段，完成特征预处理、特征筛选和模型训练和调参整个步骤。其中比较典型的 AutoML 库是 TPOT²²。其实你可以将 AutoML 看做成一个网格搜索机制：在特征工程阶段，TPOT 会尝试众多的特征编码和特征提取方式；在模型训练阶段，TPOT 也会训练多种不同类型的机器模型。

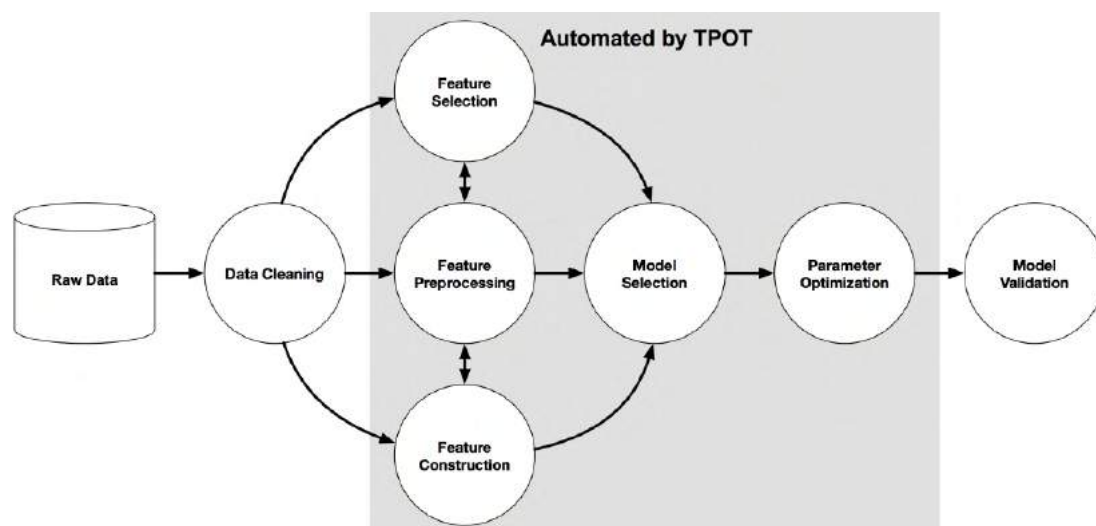


图 43: TPOP 工作流程

AutoML 还可以用于神经网络模型的设计，能够自动搜索得到较优的网络结构。在现有的深度学习网络设计的过程中，网络模型可以抽象计算模块的组合。比如在卷积神经网络中，常用的模块包括卷积层、Pooling 层、BN 层、FC 层等。2017 年谷歌利用 AutoML 的方法得到了 NASNet 网络结构，模型的计算精度和参数均优于人类最优的模型。

AutoML 的缺点是缺少人工先验知识，计算空间非常大。AutoML 设计到的算法在精度上一般很难超过人类，只能达到均值水平。综上所述，AutoML 主要有超参数筛选、自动特征工程和网络结构三个应用。AutoML 还处于快速发展的时期，暂时只能在特定场景下落地，所以不用担心失业的问题。

²¹<https://github.com/fmfn/BayesianOptimization>

²²<https://github.com/EpistasisLab/tpot>

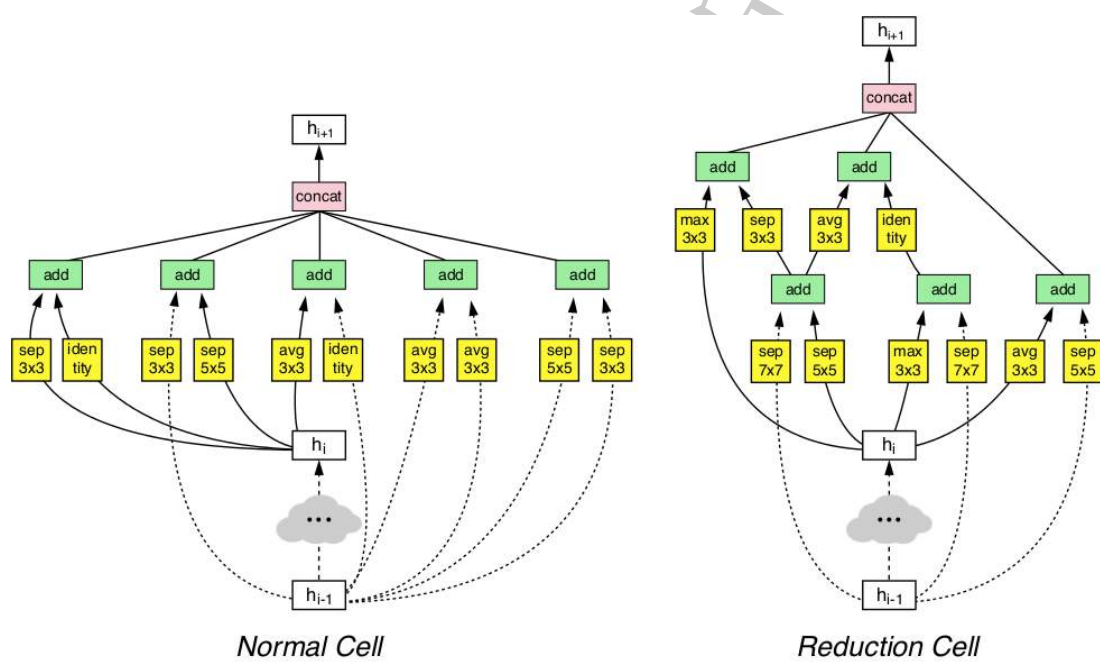


图 44: NASNet

11 侃侃而谈的博客

11.1 深度自编码器

如果给定一个神经网络，我们假设其输出与输入是相同的，然后训练调整其参数，得到每一层中的权重。自然地，我们就得到了输入 I 的几种不同表示（每一层代表一种表示），这些表示就是特征。自动编码器就是一种尽可能复现输入信号的神经网络。为了实现这种复现，自动编码器就必须捕捉可以代表输入数据的最重要的因素，就像 PCA 那样，找到可以代表原信息的主要成分。

我们将 input 输入一个 encoder 编码器，就会得到一个 code，这个 code 也就是输入的一个表示，那么我们怎么知道这个 code 表示的就是 input 呢？我们加一个 decoder 解码器，这时候 decoder 就会输出一个信息，那么如果输出的这个信息和一开始的输入信号 input 是很像的（理想情况下就是一样的），那很明显，我们就有理由相信这个 code 是靠谱的。所以，我们就通过调整 encoder 和 decoder 的参数，使得重构误差最小，这时候我们就得到了输入 input 信号的第一个表示了，也就是编码 code 了。因为是无标签数据，所以误差的来源就是直接重构后与原输入相比得到。

11.2 模型的可解释性

可解释性是非常宽泛的概念²³，但对于机器学习模型至关重要。我们不仅希望训练好一个有泛化能力的机器学习模型，还试图去理解模型的决策过程和模型参数的含义。对于自动驾驶、医疗诊断等任务而言，模型的预测结果的可解释性至关重要。可解释性强的模型精度往往不高，精度高的模型可解释差，可解释性和精度是选择模型的权衡点。

根据模型可解释性的差异，可以将机器学习模型分为黑盒模型（Black-Box Models）和白盒模型（White-Box Models）两类。黑盒模型的模型很难进行解释，模型的预测结果也难解释，但黑盒模型一般具有较高的精度。常见的黑盒模型包括神经网络和集成模型；白盒模型的参数一般都有明显的含义，模型的预测结果可解释，模型也比较简单。常见的白盒模型包括逻辑回归、线性回归、聚类算法等。

11.3 深度学习框架的选择

如果把深度学习比喻为炼丹，那么深度学习框架则可以视为炼丹炉²⁴。单炉的选择一方面可以影响模型的训练速度，还可以影响模型的精度。在现有的深度学习的『武林江湖』有几个比较大的门派，各个门派的门生以自己的刻苦修炼自己的单炉技术，希望能够练出仙丹。

与操作系统类似，深度学习框架也是互联网巨头的必争之地。亚马逊、谷歌和脸书都不留余力的推广自加的框架。本篇博客我希望给大家讲解一下我对各个深度学习框架的看法，同时也给出自己在学习和工作过程中的使用心得。希望对大家有帮助。

首先我们来看下现有的框架：

²³<https://christophm.github.io/interpretable-ml-book/>

²⁴<https://zhuanlan.zhihu.com/p/23781756>

1. **MatConvnet**: 基于 MATLAB 环境的深度学习框架, 作者为 VGG 组; 当然最初的 VGG 模型就是用这个框架训练出来的。

- 优点: 非常简单, 与 Matlab 高度集成;
- 缺点: Matlab 的版权问题, 后期框架很少维护;

2. **Theano**: 基于 MATLAB 环境的深度学习框架, 作者为 VGG 组; 当然最初的 VGG 模型就是用这个框架训练出来的。

- 优点: 非常简单, 与 Matlab 高度集成;
- 缺点: Matlab 的版权问题, 后期框架很少维护;

3. **Caffe/Caffe2**: 非常流行的计算机视觉任务, 早期大部分论文都是采用 Caffe。

- 优点: 非常适合视觉类型的任务, 框架非常清晰;
- 缺点: 不适合非视觉类型的任务;

4. **MXNet**: 亚马逊旗下的深度学习框架。

- 优点: 支持多种编程语言, 模型性能较好;
- 缺点: 官方文档比较混乱, 深入学习难度较大;

5. **TensorFlow**: Google 出品的静态图深度学习框架, Tensorflow2.0 也加入了动态图模型。

- 优点: 预训练模型以及官方文档齐全, API 丰富;
- 缺点: 不同版本代码不兼容, 学习成本较大;

6. **Keras**: 一个高层的神经网络 API 库, 底层实现可以使用 TF、Theano 和 CNTK 等框架。

- 优点: API 清晰简单, 适合入门学习;
- 缺点: 可定制性不强, 性能较差;

7. **Torch/Pytorch**: Facebook 旗下的基于动态图的深度学习框架。

- 优点: 框架灵活, 可定制性强, 社区活跃;
- 缺点: 动态图不太适合部署服务;

那么如何对框架进行选择呢, 还是要具体问题具体分析。我们这些框架无外乎是想解决特定的任务, 以下几点可能是需要考虑的:

- 优先选择有你需要的预训练模型对应的框架;
- 如果是 NLP 任务可以优先选择 Keras;
- 如果关注性能可优先选择 TensorFlow/MXNet;

对于从业者来说, 我会分析下我对深度学习框架的一些看法。

1. 首先各个深度学习框架的发现都是比较类似的：比如 TF2.0 也加入了动态图的特性，MXNet 也推出了支持动态度的 Gluon 库。所以各个框架在发展的过程中也是不断平衡框架的特性，动态度灵活性比较大，适合用来进行学习和学术研究；静态图灵活性稍差，但适合模型的部署。所以建议大家在学习的过程中，没必要将各个框架完全对立起来。
2. 选择框架应该从预训练模型入手：众所周知预训练模型是进行迁移学习的关键点，因此如果因考虑预训练模型所使用的框架；
3. 选择框架还应该从部署服务的便捷性考虑：比如 MXNet 就有编译好的经过 Intel-MKL 优化的安装包，直接 pip 安装即可。

11.4 深度学习的发展

深度学习的发展非常迅速，基本上每周都会有新的论文的出现，在国内的公众号新闻上经常看到“XX-Net 精度超过人类”的情况。这种标题党新闻的大部分内容都是：XX-Net 在某个数据集上的精度超过了人类的平均水平。我并不是批判这些论文存在的意义，而是对这种标题非常反感。这也说明了深度学习极大加速了行业的发展速度，降低了学习门槛。

在深度学习之前的时代，非结构数据的特征如何提取、特征如何表示和特征如何编码，大都是人工的领域知识。比如人脸识别的 XX 特征，物体检测的 XX 特征，这些传统算法都有很多背景知识和领域知识，想要深入学习其中的一个算法需要花费大量时间。深度学习在一定程度上降低了从业者对领域知识的限制，只要有足够的数据和定义好的模型细节，剩余的就是繁琐的实验细节了。

深度学习的魅力就在于此，在这里我不想再过多的吹捧深度学习，而是想谈论下对于我们个人而言的影响。

1. 我们如何选择模型？

但遇到一个机器学习任务之后，如何选择合适的模型呢？首先你必须清除问题的类型和数据类型，并结合现有的解决方案进行判断。毕竟我们想要的是又快又好的达到目标，知己知彼是没有坏处的。

- 问题：房价预测，模型：树模型
- 问题：图片分类，模型：CNN 模型

深度学习比较适用于非结构数据，适合解决有“语义鸿沟”的问题。当然也不是说神经网络不能做结构化数据任务，它在非结构化数据比较占优。

2. 我们如何学习深度学习？

深度学习是机器学习的分支，所以不建议把各个分支独立开来。很多深度学习最新的发展都是旧瓶装新水，饮者自知。如果你想深入学习深度学习，建议还是把机器学习的基础、深度学习的细节好好学习下，能够直接改网络结构，能够动手训练网络。

深度学习在工业界也有广泛的应用。在工业界关注的是模型的精度、模型可解释性、模型迭代速度、模型预测耗时和模型文件大小这些细节，所以在工业界更加关注深度学习的部署和迭代优化的过程。

11.5 人工智能的边界是什么？

这里我想聊聊人工智能的边界是什么？现有的人工智能本质是机器学习算法，所以机器学习算法的边界是什么？谈到边界，自然就有范围界限。所有的机器学习算法都是有适用场景的，不存在万金油的算法。

如果想应用人工智能，应该满足以下条件：

1. 具体任务明确，目标清晰；首先你必须要知道你的场景是什么、目标是什么、解决什么问题、用什么方式进行评价的，也就是首先要定义清除任务边界。如果任务都不是明确的，那么应该去明确问题的定义。

2. 模型的误差可接受；

再强大的模型都是有误差的，你不能对模型有过高的精度要求，同时模型的误差应该是在可接受范围内的。

3. 模型可反复迭代优化

机器学习模型肯定存在泛化误差，且会受到训练集和测试集分布不一致的影响。因此在模型应用之后，需要对模型进行反复优化，让模型从错误例子中再次学习。

AI时代自动驾驶是最火热的创业话题，众多的创业公司和明星大厂都成立了各自的自动驾驶部门，期望使用深度学习和机器学习技术来解决自动驾驶的问题。当然深度学习在计算机视觉技术上可以检测车辆、道路、行人和环境信息，也可以对移动的行人和车辆进行实时跟踪。



图 45: KITTI 数据集样例

大多数自动驾驶公司都是利用了基于深度学习的计算机视觉模型完成特定的任务，都声称自己的设备精度如何高，已经进行了多少里程的测试。但是截至到 2019 年底，还是很难见到有通用无人驾驶的出现。其实早在 2015 年初就有很多无人驾驶的声音，几年基本上没有成果。其中最早开始无人驾驶研究的特斯拉也基本上放弃了无人驾驶的尝试，而是把无人驾驶技术当做一种驾驶辅助功能。

所以我一直对通用常见的无人驾驶持理性态度，任务通用场景的无人驾驶短时间不可能实现，只可能在特定环境下或者作为驾驶辅助系统。无人驾驶汽车犯错人本高，迭代优化周期长，是一个非常复杂的应用系统。现有的很多无人驾驶公司都是拿着开源项目骗投资而已，站在风口上猪都可以飞起来。

11.5.1 如何选择一个合适的数据竞赛？

现在数据科学竞赛非常多，国内外大大小小的企业都可以组织各种类型的数据科学竞赛。竞赛多了可选择的机会也多了，但对于每个参赛选手来说我们的时间和精力是有限的，所以选择一个合适的竞赛参赛就至关重要了。这里的合适的含义带有一定的主观色彩，我会从竞赛内容和个人收获两个方面来阐述。

首先数据竞赛的形式非常多：有的赛题要求选手开发应用，有的要求选手设计优化算法，有的要求选手提出解决方案。选择赛题一定要选择自己擅长或者想学习的赛题，一定要选择尽量靠谱公平的赛题（国内比赛偶尔会出现名词内定的情况）。此外还要根据具体的赛题日程和规则进行衡量，尽量选择日程安排比较紧凑合理的赛题，尽量选择换排行榜（切换不同测试集重新排名）的赛题。

在课程中我们重点关注数据挖掘算法类型的数据竞赛，在此类赛题中也有不同的赛题类型，有结构化、语音、图像、视频和文本不同类型的。我期望大家是尽量能够多参加不同类型的赛题，不要受到数据形式的限制，多接触各种类型的算法和知识。其实很多知识点都是类似的，在语音识别和语音分类中的特征提取操作或许能够用在某些结构化数据上，CNN 操作也经常用在语音分类上。

此外由于数据竞赛本身具有一定的随机性（数据噪音、算法的随机性和优化过程的随机性），导致不同选手使用相同的数据和相同代码得到的结果在精度上都有差异。举个例子在 XGBoost 算法中有很多超参数可以调节，而不同的超参数可能会带来一定的精度差异。我建议大家尽量参加不是由随机性主导排名的比赛，尽量参加随机性小一点的竞赛。

最后还可以从赛题的奖励和赛制进行选择，我个人比较倾向于 TOP10 都有钱的比赛，同时国内比赛都有现场答辩的环节，所以比赛答辩城市也可以考虑下。

12 总结与展望

通过本次课程相信大家对于数据科学知识点有了一定的了解，对数据算法和具体的问题也有了一定的认识。数据科学是一个发展非常快速的领域，是一个综合性的学科，所以希望大家抱着学习和开放的心态去看问题。

当然大家在学习完本次课程后肯定对数据科学有不同的认识，在这里我想从技术和产品两个角度来谈论下数据科学。在之前章节的知识点中我们主要是从技术的角度来讲解的，在此我也希望大家能够具备一些产品思维。任何一个算法有具体的应用场景，也必须要与相应的商业环境结合；任何算法都是不断改善迭代的，并在迭代过程中分析算法是否达到了业务目标，是不是要更改业务目标。

12.1 知识点总结

12.2 就业建议

由于数据科学是一个新的学科，所以现在国内外很少有全面的培养体系。在学校我们虽然可以学习到各种知识点，我们能够知道模型的具体原理。但由于没有应用场景，我们还缺少了知识点最终应用的环节。很多学生就算读到博士毕业，可能也没有把一个项目做到工业生产的环节。

一个模型在工业界进行应用是一个非常复杂的事情，这个模型有什么依赖环境、模型的响应速度是否满足要求、模型的精度是否满足业务、模型是否可以持续优化，还有很多类似的要求都会限制模型落地。在工业界公司首要的是要考虑盈利，对于具体业务关注的是用户需求和特定应用场景，具体的算法模型可能只是其中很小的一部分。所以建议大家在学校的时候也可以培养下直接的业务思维，试着从业务和用户的角度去看一个算法模型。

最后给大家在简历上一些建议：

1. Kaggle 比赛是加分项，但并不能完全代表你的能力。之前有一位获得阿里 Fashion AI 比赛的 TOP3 选手来面试，结果决策树的知识点都不熟悉，LightGBM 的知识点也不懂。如果你想在简历上写上 Kaggle 比赛经历，一定要把写上的比赛好好吃透，把比赛的工具一定要看懂。
2. Github 最好有一些个人项目，不管项目有没有人 star。模型最好在复杂数据集上进行验证，不要只跑了 MNIST、CIFAR 这些简单的数据集，最好在大规模的数据集上进行验证过。
3. 简历上的技能最好有共同点，不相关的技能最好不要写。

13 附录

13.1 推荐书单

为了方便大家在课余时间进行学习，本节列举一些优质的书籍。这些书范围包括机器学习、数据挖掘和计算机的相关知识，也都是非常经典和有价值的资料。希望大家有空可以查漏补缺，也推荐大家如果有能力尽量阅读外文原版。

1. 《机器学习》，周志华

由周志华编写的《机器学习》又称为西瓜书，是非常经典和基础的教材。这本书的内容非常非常全面，很适合入门学习。

2. 《Deep Learning》，《深度学习》

《Deep Learning》是由 Good Fellow 等人编写的关于深度学习的教材，非常经典。这本书的内容很基础，包括的内容涵盖了深度学习的方方面面，非常适合入门深度学习。《深度学习》是国内的翻译的中文版。

3. 《Neural Networks and Deep Learning》，Michael Nielsen

这本是由 Michael Nielsen 撰写的一本关于深度学习的小手册，内容比较基础，拥有精美的绘图，也是非常基础的学习资料。

4. 《解析深度学习——卷积神经网络原理与视觉实践》，魏秀参

这本是魏秀参在博士期间撰写的一本中文卷积神经网络的教材，并且有公开的 PDF²⁵，知识点和 CS231N 公开课一致，非常适合入门学习。

5. 《动手学深度学习》，阿斯顿·张/李沐等

这本是比较新的偏实践书，对应的公开课是 STAT 157，书籍有公开的 PDF²⁶，具体实践的框架是 MXNet，也非常推荐。

6. 《神经网络与深度学习》，邱锡鹏

这本是邱锡鹏撰写的一本理论比较完整的深度学习理论教材，并且有公开 PDF²⁷，也是推荐大家学习。

7. 《用 Python 进行数据分析》，Wes McKinney

这本书内容是使用 Python 语言和相应库进行数据分析，书非常经典易懂。同时本书的作者 Wes McKinney 也是 Pandas 库的作者，可以说是非常权威了。

8. 《Learning From Data》

这本书关注于机器学习的基础概念，内容易懂。课程还有对应的公开课²⁸，值得一读。

²⁵http://lamda.nju.edu.cn/weixs/book/CNN_book.html

²⁶<https://zh.d2l.ai/>

²⁷<https://nndl.github.io/>

²⁸<https://work.caltech.edu/telecourse.html>

9. 《统计自然语言处理》

由宗成庆编写的 NLP 中文教材，适合入门学习。

10. 《Speech and Language Processing》

由 NLP 领域的大牛，斯坦福大学 Daniel Jurafsky 教授和科罗拉多大学的 James H. Martin 教授等人共同编写，想要深入学习 NLP 的同学千万不要错过。

11. 《精通数据科学：从线性回归到深度学习》，唐亘

这本书是国人撰写的数据科学的教材，知识点覆盖面非常广：从 Python 基础、计算机基础、概率论、机器学习和数据科学应用，也是给人感觉非常惊艳的一本书，适合入门学习。

12. 《百面机器学习》，诸葛越

这本书是作者在 Hulu 工作面试中总结得到的“面试书”，包括了很多机器学习的知识点，适合面试前查阅。

13.2 推荐链接

为了方便大家在学习的时候能够快速找到相应的资料，我整理用于搜索资料的的相关网站。

1. Reddit: 国外的百度百科，上面拥有众多的版块。其中比较有内容的版块有Machine Learning、Computer Vision。
2. Top arXiv papers: 将高引的 arXiv 论文进行了梳理；
3. Deep Learning Monitor: 最近的深度学习论文、推特和博客资源；
4. 专知: 会整理一些高质量的博客和论文；

13.3 推荐公开课

1. CS231N: 斯坦福开设的深度学习与计算机视觉公开课；
2. CS224: 斯坦福开设的深度学习与自然处理公开课；
3. STAT 157: Berkeley 开设的深度学习课程；
4. DS100: 数据科学技术导论；
5. CS109: 斯坦福开设的数据科学课程；
6. CMU 15-388/688: CMU 开设的数据科学课程；
7. 机器学习基石/机器学习技法: 台大林轩田开设的机器学习课程。
8. Recommender Systems Specialization: 推荐系统专项课程

13.4 常用 Python 库

1. IPython

IPython 是一个交互式的 Python 环境，还包含了 Jupyter 内核、可视化工具和并行计算工具。在 IPython 中还加入了一些魔法命令非常方便。

2. Pandas

Pandas 是基于 Numpy 的数据分析和聚合的工具，非常适合进行结构化数据的聚合、分析操作。你可以将 Pandas 视为 Excel，它能够高效的读取、处理和解析结构化表，是做数据分析必备的库²⁹。

3. Numpy

Numpy 是用于进行向量、矩阵计算的库，高效简洁。

4. SciPy

SciPy 是基于 Numpy 的一套科学计算库，提供了各种数学计算函数。

5. scikit-learn, <https://scikit-learn.org/stable/>

scikit-learn 提供了机器学习相关的函数，从预处理、特征工程、常用模型和特征筛选方法。

6. 可视化绘图库

- **Matplotlib**
- **Seaborn**
- **Bokeh**
- **Plotly**
- **folium**
- **pandas-profiling**
- **Altair**

7. tqdm

8. missingno

9. scikit-image

10. Pillow

11. OpenCV

12. XGBoost

13. LightGBM

14. CatBoost

²⁹<https://medium.com/dunder-data/minimally-sufficient-pandas-a8e67f2a2428>

15. 深度学习库

- **Keras**

Keras 是基于 TensorFlow 引擎的高层深度学习框架，可以像堆积木的方式定义网络结构。优点：简化了网络模型的细节，非常适合快速实现 demo；缺点：很难定制模型，灵活性较差。

- **TensorFlow**

Google 出品的静态图框架，Tensorflow2.0 也加入了动态图框架。优点：预训练模型以及官方文档齐全，API 丰富；缺点：API 变化较多，不同版本的代码可能无法通用。

- **MXNet**

MXNet 是一款高性能的深度学习库，用户群稍少但性能出众。优点：支持多种编程语言，框架性能较好；缺点：官方文档比较混乱，深入学习难度较大。

- **Pytorch**

Pytorch 是 Facebook 出品的动态图深度学习框架。优点：非常灵活可定制很强，非常适合构建新模型；缺点：在部署环境上还缺少相应的优化。

- **Caffe/Caffe2**

非常流行的计算机视觉任务，早期大部分论文都是采用 Caffe。Caffe2 框架以及与 Pytorch 进行合并。优点：非常适合视觉类型的任务，框架非常清晰。缺点：不适合非视觉类型的任务；

13.5 比赛经验

本节我会简单讲解下我自己的在数据竞赛中的一些实践技巧，希望能够帮助到大家。首先拿到一个赛题后，不要着急去写代码，而是要仔细收集和阅读赛题的背景。尽量去弄懂赛题的背景和要解决的问题，并寻找一些类似的解决方案。现有的数据竞赛的类型都是比较固定的，因此肯定有以及举办过的类似的赛题，而历史赛题的解决方案是非常值得借鉴的。在搜集资料的过程中，我们也可以思考下赛题的类型，可以用哪些模型进行解决。

对赛题有一定了解后就可以写代码了，可以先写一个简单的模型完成整体的流程（从数据读取、训练模型、输出结果文件），然后不断改进。同时对于特征编码和提取新特征，代码尽量高效并行。在很多比赛中，由于数量的关系导致读取数据、提取特征等操作都非常耗时间，所以建议大家还是熟悉下 Python 的多线程和多进程的操作，以及文件的存储方式。

在现有的 Pandas 库中，常见的操作比如 groupby、apply 和 map 等函数都是串行操作，并不能充分发挥多核的能力。因此我们可以手动写一些并行操作，具体的可以参考 Joblib 的例子³⁰。当然这些操作都是积累起来的，所以还是建议大家勇于尝试。

又比如文件的存储又有哪些方式呢：csv 格式、pickle 格式、numpy 矩阵格式、hdf 格式和 feather 格式，每种格式适合存什么数据、适合用在什么数据量，各自又有什么优点呢？这些都是工程细节，虽然不是特别重要，但会影响代码的读取和写入文件的速度。

³⁰<https://joblib.readthedocs.io/en/latest/>

太阳底下没有新鲜事，比赛的赛题的任务往往都是类似的。当你遇到一个新赛题后，可以找找历史有没有类似的题目，在学术界是不是有对应的领域。同类型的比赛都是类似的，也是非常值得学习和借鉴的。

13.6 常见面试题

就业是每个研发人员必须面对的问题，想要找到一份满意的工作就必须要好好准备，其中面试题是必须要准备的一项。面试的时长一般从半个小时到几个小时不等，由于面试的时间有限，面试官一般都会借助面试题来考察面试者的技术水平。面试题一般由具体的知识点构成，用来考察面试者的知识点储备、逻辑思维、专业基础和专业素养。

数据科学面试题的知识点一般由以下三类组成：

1. 专业基础知识点：计算机基础、计算机网络、计算机体系结构、数据库、操作系统和编程语言；
2. 数学基础：概率论、数理统计、矩阵计算、离散数学、数值分析和凸优化；
3. 算法与数据结构：复杂度分析、排序算法、查找算法、树算法、图算法和随机算法；
4. 机器学习：模型评估与选择、线性模型、决策树、神经网络、支持向量机、贝叶斯、集成学习、聚类和降维；
5. 深度学习：常见的模型结构、常见的网络层、优化器、训练技巧；
6. 计算机视觉：图像颜色空间、图像局部特征、图像全局特征、图像分类模型、物体检测模型；
7. 自然语言处理：语言模型、概率图模型、文本表示、句法分析、对话系统、情感分析；

数据科学相关的内容非常多，从计算机基础一致涉及到具体的专业知识，这些知识点和相关的问题是非常容易在面试中遇见的，同时也可以用自己复习查漏补缺。每个知识点的来龙去脉、知识点之间的联系，以及知识点具体细节都是值得学习和总结的。考察一个人是否入门，最直接的方法就是看他对专业知识点的理解和应用，是不是用专业术语和流程来解决问题。网上也有一些整理的较好的知识点手册³¹，也非常值得学习。

13.6.1 常规面试题

算法与数据结构

1. 单向链表的逆序输出；
2. 树的中序、先序和后序遍历；
3. 二叉搜索树/平衡二叉树/红黑树的构造过程；
4. 图的深度优先、广度优先遍历；
5. Prime 算法和 Kruskal 算法；
6. KMP 算法；
7. sqrt/atoi 的函数？

³¹<https://github.com/scutan90/DeepLearning-500-questions>

13.6.2 机器学习面试题

1. 请简述过拟合、欠拟合的定义；
2. 判别模型与生成模型有什么区别，列举几个你熟悉的生成模型；
3. 监督学习、非监督学习和半监督学习的定义，并列举分别列举几个对应的模型；
4. 请解释下维度灾难的情况；
5. 列举一个余弦相似度的应用场景；
6. 训练集/验证集/测试集有什么区别，如何进行划分的？
7. 机器学习算法如何测试，如何设计一个合理的 A/B 测试？
8. SVM 和 SVR 的公式推导
9. SVM 的软间隔问题
10. L1 正则为什么能让系数变为 0，L1 正则怎么处理 0 点不可导的情形；
11. 如何对数值特征进行归一化，有哪些归一化方法？
12. 类别特征如何编码？
13. 业务场景下如何构建特征，如何验证特征的有效性？
14. 如何对高维特征进行降维，以及有哪些特征选择的方法？
15. 线性回归与逻辑回归有什么区别和联系？
16. 最小二乘法的推导；
17. 树模型是如何选择最优的分裂节点的？
18. 树模型有哪些防止过拟合的方式？
19. 树模型比较适合何种类型的数据，分类树和回归树有什么区别？
20. 请分别叙述随机森林、XGBoost、LightGBM 的异同点；
21. 请简述 EarlyStop 的工作机制；
22. 请简述 bagging 和 boosting 的原理和应用场景；
23. 推荐系统的原理是什么，常用的推荐算法有哪些？
24. 请解释下卷积和反卷积操作的原理和用途；
25. 什么是残差网络结构，如何设计的？
26. 什么是梯度消失/梯度爆炸现象，如何避免？
27. 请简述神经网络常用的优化方法；

13.6.3 计算机视觉面试题

1. 请简述常用的图像相似度计算方法；
2. 常用的分类 CNN 模型；
3. 常见的物体检测模型；
4. 常见的语义分割模型；
5. 分类 CNN 模型的数据扩增方式；
6. 物体检测的数据扩增方式；
7. 如何提取一幅图像中的主要颜色？
8. CNN 分类网络如果如何进行多分类，网络结构怎么设计？
9. 如何在 1 亿张图像库中快速找出两张完全相同的图像？

13.6.4 自然语言处理面试题

1. 有哪些文本表示模型，各自的优缺点是哪些？
2. 常用的文本分类模型；
3. Word2Vec 的训练方法；
4. Attention 和 Transformer 结构；
5. Bert 的网络结构；

13.7 常见数据岗位

本节将会给列举几个数据相关的岗位和招聘案例，我理解的数据岗位是工作内容与数据联系紧密的岗位。比较常见的一些数据岗位包括数据分析师、机器学习工程师、数据挖掘工程师、机器学习工程师、图像算法工程师和自然语言处理工程师。

每一类岗位都有各自的侧重点，工作的内容也有较大区别。因此大家可以根据直接的背景知识和掌握的知识点来选择合适的岗位。与传统的开发岗位相比，数据岗位更加关注于数据，而不是业务逻辑。

13.7.1 数据分析师

数据分析师的工作内容自然是数据分析，数据分析是指用适当的统计分析方法对收集来的大量数据进行分析，提取有用信息和形成结论而对数据加以详细研究和概括总结的过程。在本书中也有对竞赛数据的分析，那么竞赛中的数据分析是不是就是数据分析师的工作内容呢？其实数据分析师更加关注业务逻辑，关注数据的流程和运营。

数据分析师更加侧重业务逻辑的分析，比如业务的转化率、业务用户分析、业务流程分析和业务场景分析等。数据分析师要懂业务和运营，对数据分析的工具和技巧有较高的要求。下面是一个典型的数据分析师的招聘描述：

工作内容：

1. 理解数据本质，挖掘业务场景下用户体验，驱动客户服务支持平台运营核心策略的制定和优化；
2. 基于用户行为数据，挖掘数据表的逻辑关系，发现运营中的问题，产生应用价值；
3. 负责公司运营/日志数据的统计、监控，并进行量化分析；
4. 数据分析，场景分析、用户画像特征，发现潜在的缺陷与机会，为业务决策提供数据支撑；
5. 根据数据分析结论，推导解决方案，并推进方案应用落地，驱动业务运行；

工作要求：

1. 本科以上学历，有数据分析相关工作经验；数学、统计学、计算机等相关专业优先；
2. 精通 Excel，会使用 SPSS、R、SAS 或 Python 任一软件；熟练使用 PPT 并呈现；熟练运用 SQL；
3. 拥有敏锐分析能力，严谨客观的分析态度，和用户体验有敏锐的数据洞察力；

13.7.2 数据挖掘 (DM) 工程师

数据挖掘工程师的工作内容是挖掘数据的信息，为公司的业务提供算法和数据支持。数据挖掘工程师的日常是建立模型、算法、预测等提供一些通用的解决方案。与数据分析师不同，数据挖掘工程师更加关注数据建模的过程，以及数据的业务逻辑。

数据挖掘工程师侧重数据的业务逻辑和建模，所以也要掌握一些大数据平台的数据工具，比如 Map-Reduce、Hadoop 和 Spark 等。数据挖掘工程师要有一定的业务知识和数据知识，同时也要有一定的大数据开发能力。下面是一个典型的数据挖掘工程师的招聘描述：

工作内容：

1. 负责公司相关业务的数据挖掘与分析；
2. 建立数据分析模型，提供核心算法和基础特征，为公司业务提供支持。

工作要求：

1. 熟悉 Python、Go、C++/Java 中的一种，Hadoop 相关开源组件如：hive/spark/storm 等；
2. 熟悉常见的开源组件，有大数据处理相关经验；
3. 数据挖掘、机器学习相关背景的优先；

13.7.3 机器学习 (ML) 工程师

机器学习工程师的工作内容关注于算法本身，包括阅读论文复现是算法或者改进现有的算法，并将算法进行验证和实现。机器学习工程师的日程是收集数据、整理数据、训练模型、模型调优。与数据挖掘工程师不同，机器学习工程师关注算法本身而对具体算法的业务场景关注较少。

机器学习工程师关注特定场景下算法的选择和训练，所以需要对常见的机器学习算法和深度学习算法有较深入的了解。机器学习工程师关注算法精度，所以需要有较高的数学和算法知识。下面是一个典型的机器学习工程师的招聘描述：

工作内容：

1. 大规模数据挖掘分析，从海量数据中精准定位用户需求，改进搜索相关性；

2. 负责深度语义模型、点击模型、需求识别等核心技术改进与研发；

工作要求：

1. 熟悉 Shell/Python 等至少一种脚本语言，掌握 C/C++；
2. 熟悉常用机器学习深度学习模型，有相关项目实践经验；
3. 具备搜索、文本挖掘、用户行为分析和大规模数据处理等相关领域研发经验者优先

13.7.4 图像 (CV) 算法工程师

图像算法工程师的工作内容是图像/视频相关的算法，具体来说是实现或改进图像相关的算法。与机器学习工程师类似，图像算法工程师的工作流程也是收集数据、整理数据、训练模型、模型调优，但算法和数据是图像应用场景的。

图像算法工程师关注图像算法，所以要对数字图像处理和深度学习比较了解，同时了解特定应用场景下的图像算法。针对不同的图像任务，要掌握对应的图像算法。下面是一个典型的图像算法工程师的招聘描述：

工作内容：

1. 应用先进的计算机视觉算法、统计建模，机器学习，深度学习方法建立解决实际图像问题；
2. 工作内容涉及图像检测, 物体/人脸识别, 图像分割, 图像生成, 图像/视频内容理解，图像检索等；

工作要求：

1. 精通机器视觉和数字图像处理的相关算法，如空/频域滤波、降噪、分割、配准、增强、边缘检测、形态学处理、模版匹配等；
2. 坚实的机器视觉，机器学习特别是深度学习等人工智能领域的算法基础；
3. 熟练掌握 C/C++/Python/Matlab 等编程语言。熟悉 OpenCV 等工具使用；
4. 熟悉 Caffe/Tensorflow/Torch/Mxnet 等至少一种主流框架；

13.7.5 文本 (NLP) 算法工程师

自然语言处理工程师的工作内容是文本处理的相关算法，具体来说是实现或改进文本相关的算法。与图像算法工程师相比，文本算法工程师关注文本领域的算法。

图像算法工程师关注文本算法，所以要对自然语言处理和自然语言理解比较了解，同时了解特定应用场景下的文本算法。下面是一个典型的文本算法工程师的招聘描述：

工作内容：

1. 负责自然语言处理相关底层技术和平台的算法研究与实现；
2. 负责面向搜索场景的语义分析、知识库建立、信息抽取等事宜的方法和实践；

工作要求：

1. 熟悉语法分析、句法分析、语义表示、问答系统、对话系统等应用；
2. 精通一种编程语言，如 C/C++，Java，Python 等；
3. 熟悉深度学习和常见机器学习算法的原理与算法，能熟练运用聚类、分类、回归、排序等模型；


13.8 名词中英对照表

阿水@版权所有

中文名	英文名
有监督学习	Supervised Learning
无监督学习	Unsupervised Learning
线性回归	Linear Regression
逻辑回归/对数几率回归	Logistic Regression
刀切回归	Jackknife Regression
密度估计	Density Estimation
置信区间	Confidence Interval
假设检验	Test of Hypotheses
模式识别	Pattern Recognition
时间序列分析	Time Series
决策树	Decision Trees
随机数	Random Numbers
蒙特卡洛模拟	Monte-Carlo Simulation
贝叶斯统计	Bayesian Statistics
朴素贝叶斯	Naive Bayes
主成分分析	Principal Component Analysis(PCA)
联合学习/集成学习	Ensembles Learning
神经网络	Neural Networks(NN)
深度学习	Deep Learning
支持向量机	Support Vector Machine(SVM)
聚类	Clustering
最近邻方法	Nearest Neighbors(kNN)
特征选择	Feature Selection
空间统计建模	Spatial Modeling
推荐引擎	Recommendation Engine
搜索引擎	Search Engine
归因模型	Attribution Modeling
协同过滤	Collaborative Filtering
规则系统	Rule System
连锁分析	Linkage Analysis
关联规则	Association Rules
打分引擎	Scoring Engine
预测建模	Predictive Modeling
博弈论	Game Theory
数据填充	Imputation
生存分析	Survival Analysis
统计套利	Statistical Arbitrage
产量优化	Yield Optimization
交叉验证	Cross Validation
模型拟合	Model Fitting
关联算法	Relevancy Algorithm
实验设计	Experimental Design

数据科学 & 数据竞赛

THINK OUTSIDE THE BOX



人生的本质，
是在不断过拟合与欠拟合
by 阿水