机器学习介绍

钱坤

(中国地质大学（武汉）数学与物理学院 武汉 430074)

摘 要 自计算机被发明时起, 人们便思考能否让机器学会自主思考, 针对该问题的深入研究使得诞生了“机器学习”这一学科, 并在近些年取得巨大进步及广泛应用. 本质上, 机器学习是一个多学科的领域, 吸取了概率统计、智能优化、神经科学、信息论、控制论乃至哲学等学科的成果. 本文首先介绍机器学习的发展历程, 接着介绍了各研究方向及现状, 最后重点介绍各个方向的经典算法.

关键词 机器学习；监督学习；无监督学习；半监督学习

Research and application of arithmetic coding

Qian Kun

(School of Mathematics and Physics, China University of Geosciences, Wuhan 430074)

**Abstract** Since the invention of computer, Thinking about whether machines can learn to think independently has been starting. The in-depth research on this problem has led to the birth of the subject of "machine learning", which has made great progress and wide application in recent years. Essentially, machine learning is a multidisciplinary field, drawing on the results of probability statistics, intelligent optimization, neuroscience, information theory, cybernetics and even philosophy. This paper first introduces the development of machine learning, then introduces the research direction and the status quo, and finally focuses on the classical algorithms in each direction.

**Key words** Machine learning, Deep learning

# **机器学习发展历程**

机器学习是人工智能发展到一定时期的必然产物, 最早可追溯到对人工神经网络的研究, 提前于1946年世界上第一台电子计算机ENIAC的发明. 1943年, Warren McCulloch 和 Walter Pitts 提出了神经网络层次结构模型, 为机器学习的发展奠定了基础. 不久后, “人工智能之父”图灵于1950年发表了《Computing Machinery and Intelligence》并提出著名的“图灵测试”, 后于1956年以《Can Machine Think?》为题重新发表该论文, 其中提到了机器学习的可能, 为后来的人工智能科学提供了开创性的构思.

机器学习的相关研究于二十世纪五十年代起便迅速展开, 例如1959年年美国IBM公司的A. M. Samuel设计了一个具有学习能力的跳棋程序. 曾战胜了美国一个保持 8 年不败的冠军. 二十世纪五十年代中后期, 基于神经网络的“连接主义”学习开始出现——1957 年, 康内尔大学教授 Frank Rosenblatt设计出了第一个计算机神经网络, 成为神经网络模型的开山鼻祖. 此后, 在六七十年代出现的基于逻辑表示的“符号主义”学习技术蓬勃发展, 代表性的工作有P. Winston的“结构学习系统”、R. S. Michalski等人的“基于逻辑的归纳学习系统”等；同时, 以决策理论为基础的学习技术及强化学习技术也得到了一定的发展；此外, 二十年后红极一时的统计学习理论的一些奠基性工作也于这个时期取得.

从二十世纪五十年代到七十年代初, 人工智能研究处于“推理期”, 人们认为只要赋予机器逻辑推理能力便可让其具有智能. 代表性的工作有A. Newell 和H. Simon设计的“逻辑推理家”程序, 于1952年证明了数学家罗素和怀特海的名著《数学原理》中的38条定理, 并于1963年证明全部的52条定理. 那个时期, 机器学习作为一新生学科取得了许多重要成就, 诸如深度学习、强化学习、决策学习等现代机器学习算法的基本工作皆在该时期奠定；其中, 神经网络算法在实际问题上取得了较好成绩, 但Marvin Minsky和 Seymour Papert于1969年提出的XOR问题将其送入了不归路, 此后基于神经网络的机器学习研究陷入了十几年的低潮期.

从二十世纪七十年代中期开始, 人工智能研究进入了“知识期”. 1965年, “知识工程之父”Feigenbaum主持研制了世界上第一个专家系统DENDRAL, 此后诞生了大量专家系统, 并在很多应用领域取得了大量成果. 但是, 专家系统发展过程中面临“知识工程瓶颈”, 即由人将知识总结出来传授给计算机是相当困难的, 于是“如何让机器自己学习知识”这一思考促使了机器学习的复兴. 1980年夏, 卡内基·梅隆大学举行了第一届机器学习国际研讨会；1986年, 第一本机器学习专业期刊《Machine Learning》创刊；1989年, 人工智能领域权威期刊《Artificial Intelligence》出版机器学习专辑；1990年, MIT出版了《机器学习：范型与方法》. 总的来看, 二十世纪八十年代起, 机器学习成为一个独立的学科领域, 各种机器学习技术得到迅速发展.

1983年, R. S. Michalski等人将机器学习研究划分为“从样例中学习”、“在问题求解和规划中学习”、“通过观察和发现学习”、“从指令中学习”等种类. 同年, E. A. Feigenbaum等人在著名的《人工智能手册》（第三卷）中, 将机器学习划分为“机械学习”、“示教学习”、“类比学习”和“归纳学习”. 二十世纪八十年代以来, 机器学习领域最主要的研究为“从样例中学习”（也即广义的归纳学习）, 涵盖了监督学习、无监督学习等.

在二十世纪八十年代, “从样例中学习”的一大主流为符号主义学习, 其代表包括决策树及基于逻辑的学习. 1979年, J. R. Quinlan提出了著名的ID3算法, 此后又进一步改进提出了著名的C4.5算法.

二十世纪九十年代中期前, “从样例中学习”的另一大主流为基于神经网络的连接主义学习. 1983年, Hopfield提出了Hopfield网络, 在求解旅行商问题这一著名NP难题上取得了重大进展. 1986年, Hinton、Rumelhart、Williams发明了著名的BP算法, 产生深远影响. 1989年, LeCun提出了LeNet-5模型, 作为目前最流行的卷积神经网络（CNN）的雏形, 成功应用于手写数字识别, 该模型也是第一个被成功训练的人工神经网络.

进入二十世纪九十年代中期, “统计学习”诞生, 其包含的多种浅层机器学习模型相继问世, 代表性的技术有逻辑回归、支持向量机以及更一般的“核方法”等. 这些机器学习算法的共性是数学模型为凸代价函数的最优化问题, 理论分析简单且易于训练, 因此迅速成为主流. 但由于有限的样本及计算单元, 使得模型只能提取数据的初级特征, 不具备较强的学习能力.

进入二十一世纪后, 连接主义学习以“深度学习”为名卷土重来, 并迅速成为目前的主流. 深度学习模型, 通常为具有多个隐藏层的人工神经网络, 有良好的特征学习能力, 但模型本身复杂度极大, 该技术兴起的根本原因在于计算能力的迅速提升以及数据量的快速增多. 虽然深度学习技术缺乏严格的理论基础, 但降低了机器学习应用的难度, 只需要肯花功夫“调参”, 模型就能取得较好效果. 因此, 深度学习技术在图像、语音、文本、机器人控制等复杂对象的应用中取得了优越性能, 例如谷歌翻译, 苹果的语音助手Siri、微软的语音助手Cortana, 以及前年的AlphaGo围棋程序等.

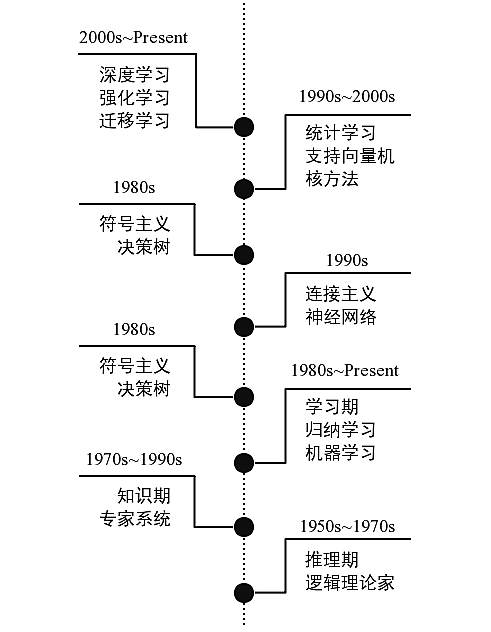


图1 机器学习发展历程

**Fig.**1 **The development of Machine learning**

# **机器学习算法**

我们知道, 机器学习算法是一种能让机器从数据中学习的算法, 由此需要给“学习”一个准确的定义. 1997年, Mitchell提出了简洁的定义：“对于某个任务*T*和性能度量*P*, 一个计算机程序被认为可以从经验*E*中学习是指, 通过经验*E*改进后, 它在任务*T*上由性能度量*P*衡量的性能有所提升. ”可以看出, 机器学习算法由任务*T*、经验*E*和性能度量*P*构建而成, 下文将分别从这三个方面对其进行阐述.

## 任务*T*

针对常规具有明确流程的任务所编写的确定性程序并不能称之为机器学习, 机器学习通常是用于帮助人们解决一些人为设定以及确定性程序难以解决的问题.

学习本身是一个获得完成任务能力的过程, 通常机器学习任务定义为机器学习系统应如何处理样本, 而样本是包含各类已量化特征的需要处理的对象或事件. 机器可以通过已有的样本, 掌握对这些或未知的样本进行处理的能力, 以此完成某些任务.

机器学习可以解决诸多类型的任务, 常见的任务类型列举如下:

* **分类**：对于某具有*n*个特征的样本, 计算机程序需要指定该样本属于*k*类中的哪一类. 学习算法通常会返回一个函数. 设, 代表算法将向量映射到数值所代表的类别. 例如在手写数字识别任务中, 机器需要学到将所给手写数字图片分类到中某一数字. 另一著名应用为ImageNet项目, 有超过1400万张带标注图片, 机器可利用这些数据学习分类1000种对象以实现图像识别.
* **回归**：在该任务中, 计算机需要根据已知输入返回预测值. 学习算法通常会返回一个函数. 这一类任务的事例有, 房价预测、股票预测等, 这类预测常用于算法交易中. 对比分类问题, 两者的区别在于回归任务的输出为连续值, 而分类任务的输出为离散值. 事实上, 回归任务可以用来解决分类任务, 常规分类任务的实际输出值为样本归属各类的概率, 再选取概率值最大的类作为其分类结果.
* **转录**：在这类任务中, 机器学习系统的输入为一些非结构化的数据, 如文本图片、语音等, 并将其转录为离散文本信息. 例如文字识别, 将文本图片中的信息转录为文字输出；语音识别, 将音频信号转录为所说的字符或单词ID的编码.
* **机器翻译**：机器翻译任务, 本质上是将输入的某种语言的符号序列, 转化成另一语言的符号序列. 通常应用于自然语言处理中, 例如科大讯飞的人工智能同声传译软件, 网易的有道翻译官等.
* **结构化输出**：结构化输出的结果为向量或是包含多个值的数据结构, 且输出的不同元素间有重要联系. 事实上, 转录、翻译等任务都可以归属于这一范畴内. 其他的应用, 例如语法分析——将自然语言句子映射到语法结构树, 并标记树的节点为动词、名词、副词等. 此外, 还有图像描述, 对输入的一幅图像运用自然语言句子描述该图像.
* **异常检测**：此类任务类似于分类任务, 计算机程序在一组事件或对象中筛选, 找出并标记非正常或非典型个体. 例如, 用户的信用卡发生非正常的交易行为, 公司可紧急冻结该信用卡, 避免可能因被盗窃而产生的财产损失.

上述介绍仅为部分机器学习可完成的任务, 并非严格意义上的机器学习任务分类.

## 性能度量*P*

性能度量是一种评估机器学习算法性能的定量度量. 需要注意的是, 性能度量*P*通常是依据任务*T*所制定的.

通常, 我们更关注机器学习算法在未观测数据上的性能, 这决定了在实际应用中的性能. 因此, 区别于已观测过的“训练集”, 我们用未观测过的“测试集”对模型性能进行评估.

### 准确率

在分类及转录任务中, 常采用“准确率”作为性能度量. 机器学习算法*f*在样例集*D*上的分类准确率定义为

## 查准率、查全率与*F*1

在某些场合, 例如流行病的检测中, 查准或查全往往更总要, 因此有时需采用“查准率”或“查全率”作为性能度量, 给定分类结果的“混淆矩阵”如下表所示

表1 分类结果混淆矩阵

**Table** 1 **Classification results confusion matrix**

|  |  |  |
| --- | --- | --- |
| 真实情况 | 预测结果 | |
| 正例 | 反例 |
| 正例 | *TP* (真正例) | *FN* (假反例) |
| 反例 | *FP* (假正例) | *TN* (真反例) |

查准率*P*与查全率*R*分别定义为

可以看出查准率和查全率彼此矛盾, 而实际问题中比较两种机器学习算法的性能, 需要同时考虑查准率和查全率, 我们常用*F*1度量作为综合考虑这两者的性能度量

在实际应用中, 有时更重视查准率, 有时更重视查全率, 因此给出*F*1度量的一般形式——, 定义如下

## ROC与AUC

通常, 很多分类器对测试样本预测出一个之间的实数, 然后将此值与0.5比较, 大于0.5则判为正例, 否则为反例. 本质上说, 这一预测值等价于该样本为正例的概率值, 我们可以依据此预测值将测试样本进行排序, “最可能”为正例的排在最前面, 以此类推, 得到新的样本序列.

针对这个样本序列, 可以逐个设置“截断点”将样本分为两个部分, 前一部分判作正例, 后一部分判作反例. 每次计算“真正例率”(TPR) 和“假正例率”(FPR), 两者定义如下

以假正例率为*x*轴, 真正例率为*y*轴, 依次连接每次计算结果得到“ROC曲线”, 以鸢尾花分类为例绘制出ROC曲线如下图所示.

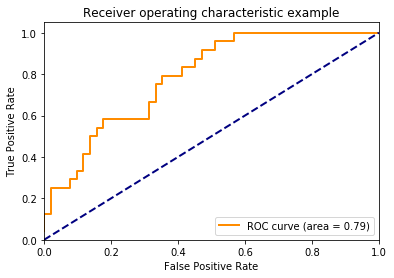


图2 ROC曲线示例图

**Fig.2 ROC curve example diagram**

得到的ROC曲线并不能直接反映机器学习算法的性能好坏, 但可通过计算ROC曲线下面积作为判断依据, 记为AUC. 由图可知, AUC值越大, 则模型分类性能越好, 计算公式如下

## 经验*E*

目前, 大部分机器学习算法是从数据上获取经验. 根据学习过程的不同经验, 机器学习算法可以大致分为无监督学习、监督学习、半监督学习、迁移学习和强化学习.

### 监督学习

监督学习算法的数据集通常含有多个特征, 其本质特征在于每个样本都有一个“标签”或“目标”. 机器通过比较预测结果与真实结果对算法进行调整, 此时“标签”或“目标”发挥着“监督”机器学习过程的作用, 故称之为“监督学习”.

监督学习的典型算法有：线性回归、逻辑回归、多层感知机、卷积神经网络等；典型应用有：回归分析、分类等.

### 无监督学习

与监督学习相反, 无监督学习算法的的数据集不显式地有“标签”或“目标”, 仅含有多个特征. 算法希望通过学习寻求数据间的内在模式和统计规律, 从而获得样本数据的结构特征. 需要注意的是, 无监督学习往往不需要事先进行训练. 这种“无中生有”的思想更接近人类的学习方式.

无监督学习的典型算法有：自动编码机、受限玻尔兹曼机, K-means等；典型应用有：聚类和异常检测等.

### 半监督学习

半监督学习算法的数据集结合了前两者算法的数据集, 即部分数据集有“标签”或“目标”, 而另一部分数据集没有. 在现实任务中, 未标记样本多、有标记样本少是一个比较普遍的现象. 算法希望可利用好大量未标记数据及少量已标记数据提升模型泛化能力. 半监督学习算法需假设未标记样本所揭示的数据分布信息与类别标记存在联系, 一种是聚类假设而另一种为流形假设.

半监督学习的典型算法有：直推式SVM、图半监督学习、协同训练等；典型应用涵盖了监督学习与无监督学习的应用范畴.

### 迁移学习

迁移学习算法类似于半监督学习, 同样是利用某一仅有少量标签样本的数据集进行学习, 区别在于后者是运用已有的模型来解决此缺乏标签样本数据的学习问题. 通俗地讲, 一个人学会了骑自行车, 则他很容易学会开摩托车. 在实际应用中, 例如训练一个识别猫的神经网络, 若从头开始训练则需要百万张带标注的数据及大量的显卡资源. 而利用迁移学习, 可使用VGG16这样成熟的模型, 只需用少量数据训练最后一层, 即可快速得到较好的识别模型.

### 强化学习

强化学习算法与前四者最大的区别在于, 机器的学习过程并不局限于一个固定的数据集上, 而会与“环境”进行交互.

机器在于环境交互的过程中, 其行为会获得一定的奖惩. 机器需要通过在环境中不断尝试而学得一个“策略”, 使得长期累积的奖赏最大化.

目前强化学习技术在游戏、机器人控制、参数优化、机器视觉等领域中得到了广泛的应用, 并被认为是迈向通用人工智能的重要途径.

# **常用算法简介**

接下来我们将介绍八个常用的机器学习算法.

## C4.5

C4.5作为一种决策树算法, 前身是著名的ID3算法, 二者的区别在于划分属性的准则.

决策树是一类常用的机器学习算法, 顾名思义, 决策树是基于树结构进行决策的, 这一点恰是人类面对决策问题的处理机制. 算法通过把实例从根结点排列到某个叶子结点实现分类, 其核心问题在于选择分裂属性和决策树的剪枝. 常用的算法有ID3、C4.5、CART等, 这些算法均采用自顶而下的贪婪算法, 每个结点选择分类效果最好的属性将结点分裂成2个或多个子结点, 下图为根据天气情况判断“周六是否适合打网球”的决策树模型.

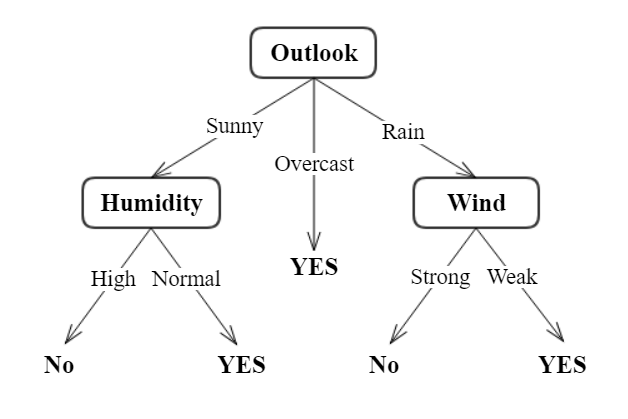


图3 决策树示例图

**Fig.**3 **Decision tree example diagram**

一般的, 一棵决策树包含一个根结点、若干个内部结点和若干个叶结点, 叶结点对应着决策结果, 其他每个结点对应一个属性测试. 由根结点到叶结点的过程即为决策过程, 决策树学习的目的是为产生一种泛化能力较强的分类模型. 通常先选取一个根结点, 根绝其属性生成各个分支, 再以每个分支为根结点或叶结点进一步划分, 具体流程如下

|  |
| --- |
| **输入**：训练集  属性集  **过程**：函数TreeGenerate  1：生成结点node；  2：**if** *D*中样本全属于同一类别C **then**  3: 将node标记为*C*类叶结点；**return**  4：**end if**  5：**if**  **OR** *D*中样本在*A*上取值相同**then**  6： 将node标记为叶结点, 其类别标记为*D*中样本数最多的类；**return**  7：**end if**  8：从*A*中选择最优划分属性  9：**for** 的每一个值**do**  10： 为node生成一个分支；令表示中在上取值为的样本子集  11： **if** 为空 **then**  12： 将分支结点标记为叶结点, 其类别标记为中样本最多的类；**return**  13： **else**  14： 以TreeGenerate  15： **end if**  16：**end for**  **输出：**以node为根结点的一棵决策树 |

图4 决策树学习基本算法

**Fig.**4 **Basic learning algorithm of decision tree**

决策树的生成是一个递归过程, 有三种情形会导致递归返回：①当前结点包含的样本全属于同一类别, 无需划分；②当前属性即为空, 或是所有样本在所有属性上的取值相同, 无法划分；③当前结点包含的样本集合为空, 不能划分.

对于第②种, 直接将当前结点标记为叶结点, 且类别设定为该结点所包含样本最多的类别；对于第③种, 将当前结点标记为叶结点, 将其类别设定为其父结点所包含样本最多的类别.

算法最核心的部分为第8行选择最优划分属性, 不同的划分方法对应着不同的决策树算法.

我们先给出“信息熵”的定义：

假定当前样本集合*D*中第*k*类样本所占的比例为, 则*D*的信息熵定义为

若使用某离散属性*a*对样本集*D*进行划分, 设其有*V*个可能的取值, 则会产生*V*个分支结点, 第*v*个分支结点包含了*D*中所有属性*a*取值为的样本, 即为. 由此可计算的信息熵, 考虑各分支结点样本数量不同, 赋予第*v*个分支结点权重, 于是可计算用属性*a*对样本进行划分所得到的“信息增益”如下

若在算法流程第8行选取使信息增益最大的属性作为最优划分属性, 即可得到ID3决策树学习算法.

由于信息增益偏好取之数目较多的属性, 因此可考虑采用“增益率”来选择最优划分属性, 以减少此偏好带来的不利影响. 增益率定义如下

其中

称为属性*a*的“固有值”

C4.5算法不直接采用增益率最大的划分属性, 而是从候选划分属性中找出信息增益高于平均水平的属性, 再从中选择增益率最高的.

## *k*-Means

在先前介绍的“无监督学习”方法中, 研究最多、应用最广的一大领域是“聚类”. 聚类试图将数据集划分为若干个互不相交的子集, 每个子集称为一个“簇”. 聚类算法可以解释数据的内在性质及规律, 获取数据的一些潜在的概念, 需注意这样的概念是事先未知的, 在聚类完成后需要由使用者把握及解释.

通俗地讲, 聚类是将相似的事物聚集在一起, 而将不相似的事物划分到不同的类别的过程. 那么, 如何衡量两个事物之间的相似度, 我们可通过定义“样本间距离”及“类间距离”进行衡量.

首先给定两样本与, 常用的“距离度量”有

1. **欧氏距离**
2. **曼哈顿距离**
3. **切比雪夫距离**
4. **闵可夫斯基距离**

* 时, 是曼哈顿距离；
* 时, 是欧式距离；
* 时, 是切比雪夫距离.

1. **马氏距离**

有*M*个样本向量, 协方差矩阵记为, 均值记为向量, 则其中样本向量到的马氏距离表示为：

1. **汉明距离**

两个等长字符串与之间的汉明距离定义为两个（相同长度）字对应位不同的数量. 例如字符串“1111”与“1001”之间的汉明距离为2.

1. **Pearson相关系数**
2. **夹角余弦**

需要说明的是, 用距离作为亲疏程度的度量值时, 距离越小, 样品之间的关联性越大；用相似系数作为亲疏程度的度量值时, 相似系数的绝对值越大, 意味着指标之间的关联性越大.

*k-*Means算法需事先给定聚类个数*k*, 聚类目标为最小化*m*个数据与其对应聚类中心点距离平方和

其中

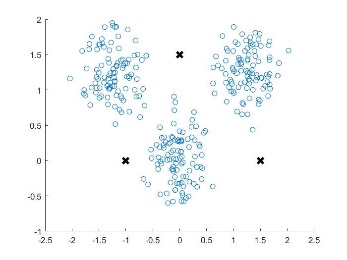
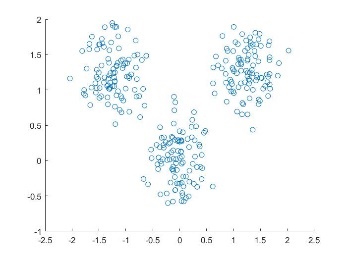
数学上已经证明这是一个NP难问题, 因此*k-*Means算法采用了贪心策略, 通过迭代优化来近似求解式(3.12), 具体流程如下

|  |
| --- |
| **输入**：样本集  聚类簇数*k.*  **过程**：  1：从样本集*D*中随机选择*k*个样本作为初始均指向量  2：**repeat**  3: 令  4： **fordo**  5： 计算样本与各均值向量的距离：;  6： 根据到最近的均指向量确定的簇标记：  7： 将样本划入相应的簇：;  8： **end for**  9： **for** **do**  10： 计算新均值向量：  11： **if then**  12： 将当前均值向量更新为  13： **else**  14： 保持当前均值向量不变  15： **end if**  16： **end for**  17：**until** 当前均值向量均未更新  **输出：**簇划分 |

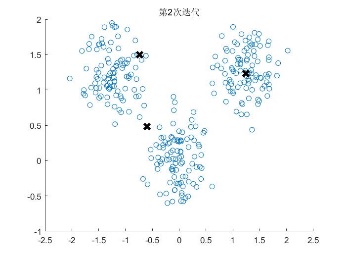
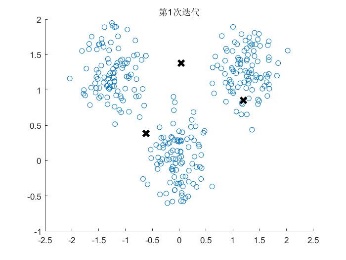
图5 *k*-Means算法

**Fig.**5 ***k*-Means algorithm**

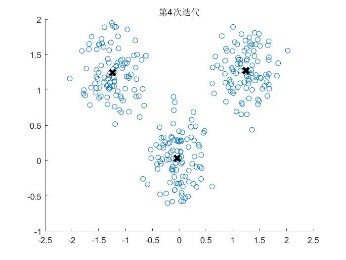
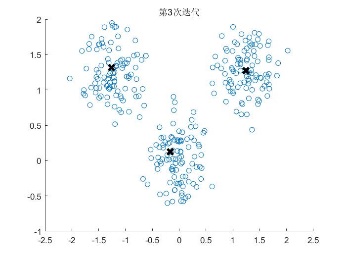
下面演示了*k-*Means算法的迭代过程：



|  |  |
| --- | --- |
| 图6 原始数据分布图 | 图7 初始化聚类中心 |
| **Fig.**6 **Original data distribution diagram** | **Fig.**7 **Initialize the clustering centers** |



|  |  |
| --- | --- |
| 图8 第一次迭代结果 | 图9 第二次迭代结果 |
| **Fig.**8 **The first iteration result** | **Fig.**9 **The second iteration result** |



|  |  |
| --- | --- |
| 图10 第一次迭代结果 | 图11 第二次迭代结果 |
| **Fig.**10 **The third iteration result** | **Fig.**11 **The fourth iteration result** |

## *k*NN

*k*NN算法是一种常用的监督学习算法, 算法工作机制非常简单：给定测试集, 基于某种距离度量找出与训练样本最近的*k*个样本, 基于这*k*个“邻居”的信息来进行预测. 通常, 在分类任务中, 可采用“投票法”, 即选取这*k*个样本中出现最多的类别作为预测结果；在回归任务中, 可采用“平均法”, 即将这*k*个样本的输出平均值作为预测结果；此外, 还可以根据距离远近做加权投票或加权平均, 距离越近权重越大.

同*k-*Means, *k*NN算法有两个重要参数, 一是参数*k*, 当*k*取不同值的时候, 分类结果会显著不同, 另一个是距离计算方式, 若采用不同的距离度量, 可能找到的*k*个“近邻”会有显著差别, 导致分类结果的显著不同.

*k*NN算法是一类典型的“懒惰学习”算法, 即在训练阶段仅保存样本, 训练时间开销为零, 待收到测试样本后再进行处理；与之对应的, 例如上述两个学习方法, 都属于“急切学习”的范畴.

## SVM

SVM, 全称为“支持向量机”(Support Vectory Machine), 是统计学习中一类非常重要的机器学习算法, 具有较好的适应能力及较高的区分率.

给定训练数据集, 找到一个直接划分两个不同的样本类的超平面是一个自然的思路. 假设存在这样一个超平面划分两个样本类, 则称这一问题为“线性可分的”, 但这样的超平面可能有许多个, 如下图, 如何找到一个“最合理”的超平面是我们需要思考的问题.

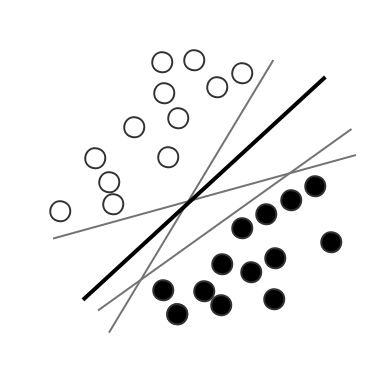


图12 存在多个超平面可将两类样本划分开

**Fig.**12 **Multiple hyperplanes can divide two kinds of samples**

直观上, 我们希望找到一个位于两类训练样本“正中间”的超平面, 即到两类样本距离较为接近的超平面, 如上图纯黑色的直线. 这样的划分超平面对噪声的“容忍性”最强, 即具有最好的泛化性能.

通常, 一个划分超平面可表示为一线性方程：

其中为法向量, 决定超平面的方向；参数*b*为位移量, 决定超平面到原点距离, 可得任意点到该超平面的距离为：

假设该超平面可正确划分训练样本集, 且对任一样本对, 该超平面满足

且距离该超平面最近的样本点满足使等号成立, 这样的样本点被称为“支持向量”. 根据分析可知这样的和总是存在的, 且有两个异类支持向量到超平面的距离和为

称之为“间隔”.

则“线性可分支持向量机”算法的学习目的为找到具有“最大间隔”的划分超平面, 即

等价于

此为SVM的基本型, 该问题为等式约束的图二次规划问题, 可采用拉格朗日乘子法求解, 可写出拉格朗日函数为

其中, , 令对和的偏导为零可得

将式(3.22)代入(3.21)得(3.20)的“对偶问题”如下

注意到式(3.20)有不等式约束, 可知上述过程需满足KKT条件, 即要求

于是, 由第三式可知, 对任意训练样本, 总有或. 若, 则在式(3.22)中不会出现；若, 则必有, 该样本为支持向量. 因此最终模型只与支持向量有关, 故模型称之为“支持向量机”.

式(3.24)为一个二次规划问题, 可采用通用的二次规划算法求解, 问题规模正比于训练样本数. 若考虑问题本身的特性, 则存在一些高效的求解算法如SMO等.

若某样本集不存在线性划分超平面, 例如“异或”, 则上述的“线性可分支持向量机”模型无法求解. 幸运的是, 如果原始空间为有限维（属性有限）, 则必存在一个非线性映射将原样本映射到一高维特征空间且满足样本集线性可分.

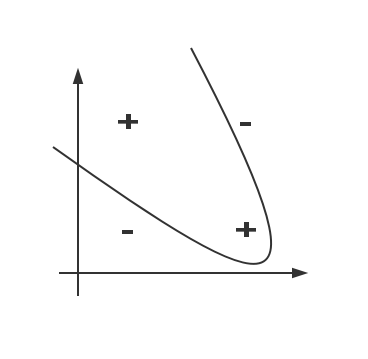


图13 异或问题

**Fig.**13 **XOR problem**

原模型可表示为

其中和为模型参数, 类似的有

对偶问题为

注意到式(3.28)中为样本与样本映射到特征空间后的内积, 有时直接计算很困难, 可设想一个函数：

式(3.28)可改写为

求解得

该方法称之为“核方法”. 由于不知道特征映射的形式, 无法确定选择哪个核函数是最合适的, 因此核函数的选择也是一重要变量.

下表列出了常用的核函数：

表2 常用核函数

**Table** 2 **Common kernel function**

|  |  |  |
| --- | --- | --- |
| 名称 | 表达式 | 参数 |
| 线性核 |  |  |
| 多项式核 |  |  |
| 高斯核 |  |  |
| 拉普拉斯核 |  |  |
| Sigmoid核 |  |  |

## Naive Bayes

假设共有*N*个可能的分类分类, 即记作是将一个真实标记为的样本分类为的损失, 由此基于后验概率我们可以计算出样本被分类为的期望损失, 即在样本上的条件风险

我们希望找到一个判定准则以最小化总体风险

由此定义贝叶斯判定准则：对每个样本, 选择可使条件风险最小的类别标记, 即可最小化总体风险, 称该判别准则为贝叶斯最优分类器

若分类的最优目标为最小化分类错误率, 则判别损失可写作

此时条件风险

最小化分类错误率的贝叶斯最优分类器为

由贝叶斯定理知, 可写作

基于贝叶斯公式估计后验概率有一大难点为类条件概率是所有属性上的联合概率, 难以从有限的样本中直接估计, 故提出“属性条件独立性假设”, 即对已知类别, 假设所有属性相互独立, 式(3.38)可写作

其中*d*为属性数目, 为在第*i*个属性上的取值.

对所有类别来说相同, 则贝叶斯最优分类器可写作

此即朴素贝叶斯分类器的表达式. 因此, 朴素贝叶斯分类器的训练过程需要基于训练集*D*来估计类先验概率, 并为每个属性估计其条件概率.

令为训练集中类别为第类的全部样本, 可估算出类先验概率

若属性为离散属性, 令表示中在第*i*个属性上的取值为的样本组成的集合, 则条件概率可估计为

对于连续属性, 可假设服从正态分布, 即假定和分别为*c*类样本在第*i*个属性上取值的均值与方差, 条件概率密度可估计为

## BP

日前最热门的机器学习研究领域为深度学习方向, 其基础为神经网络算法. 前文介绍到神经网络于二十世纪五十年代便诞生萌芽, 而机器学习真正迈入“深度学习”时代源于2006年Hinton提出并解决了训练过程的深度置信网络(DBN), 成名于2012年ImageNet比赛利用AlexNet以高于第二名十几个百分点的准确率夺得冠军.

首先介绍最基本的神经元模型. 考虑生物神经元, 每个神经元与若干其他神经元连接, 当它“兴奋”时则向其他神经元传递化学物质, 并改变这些神经元的电位, 若某个神经元电位超过一定“阈值”, 则被激活, 即“兴奋”起来, 向其他神经元传递信息. McCulloch于1943年提出的“M-P神经元模型”便是抽象了上述情形, 该模型中, 神经元接受*n*个其他神经元传递来的信号, 并采用加权和作为总输入值, 若大于阈值则被激活并通过“激活函数”处理后输出.

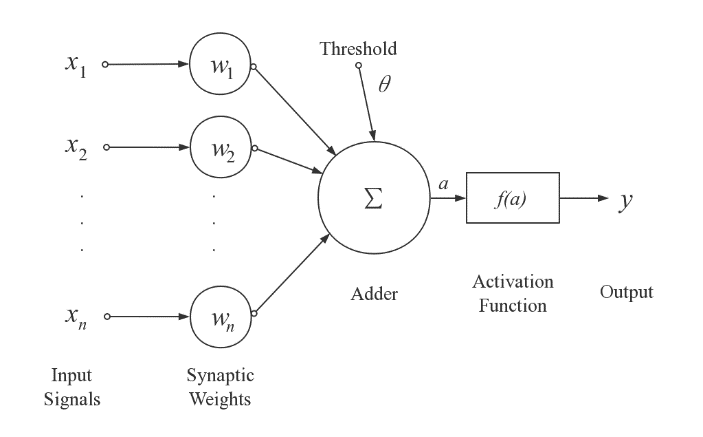


图14 M-P神经元模型

**Fig.**14 **M-P neuron model**

模型需给定激活函数, 常用的有Sigmoid函数等, 模型的训练过程就是调整模型内部参数, 包括权值和阈值.

进一步的, 将许多个这样的神经元按一定的层次结构连接起来, 就得到了神经网络

最简单的神经网络模型为感知机, 可实现逻辑与、或、非运算. 该模型由两层神经网络组成, 输入层接收外界输入信号后传递给输出层, 输出层为M-P神经元. 对于有个输入单输出的感知机, 其输出可表示为, 可将阈值视为输入固定为的结点所对应的权重.

设激活函数为单位阶跃函数, 给定训练样本, 设其输出为, 则感知机的权重可以如此调整：

其中称为学习率, 实际是输出关于参数的偏导数值

对于大于两层的神经网络, 规定第一层为输入层, 最后一层为输出层, 其余为隐含层. 称具有隐含层的网络为“多层网络”, 利用多层网络学习便称为“深度学习”.

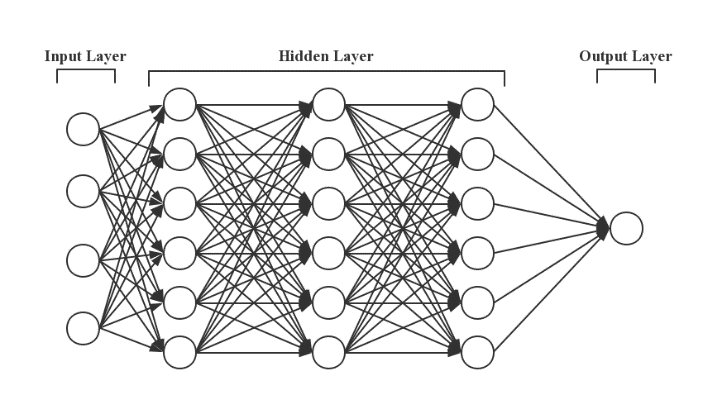


图15 多层神经网络

**Fig.**15 **Multi-layer neural network**

多层网络的学习能力要远强于单层感知机, 而训练难度也远复杂于单层感知机, 直到误差逆传播(简称BP)算法的出现才解决了这一问题. 给定一多层神经网络, 设激活函数为Sigmoid函数, 输出误差为均方误差, 算法流程如下

|  |
| --- |
| **输入**：样本集  学习率  **过程**：  1：在范围内随机初始化网络中所有连接权和阈值  2：**repeat**  3： **for all**  **do**  4：计算每个隐含层结点的输入：  5：计算每个隐含层结点的输出：  6： 计算每个输出层结点的输入：  7： 计算每个输出层结点的输出：  8： 计算每个输出层结点的误差：  9： 计算每个隐含层结点的误差：  10： 更新输入层到隐含层的权值：  11： 更新隐含层到输出层的权值：  12： 更新每个隐含层结点的偏置：  13： 更新每个输出层结点的偏置：  14：**until** 满足终止条件  **输出：**训练完成的多层神经网络 |

图16 BP算法

**Fig.**16 **Error back propagation algorithm**

## CNN

CNN为目前应用最广泛的神经网络模型. 1962年, Hubel和Wiesel通过对猫视觉皮层细胞的研究(局部敏感和方向选择的神经元具有独特的网络结构), 提出了“感受野”(receptive field)的概念. 感受野以某种方式覆盖整个视觉域, 它在输入空间中起局部作用, 因而能够更好地挖掘出存在于自然图像中强烈的局部空间相关性.

1984年, 日本学者Fukushima基于感受野概念提出的神经认知机, 是感受野概念在人工神经网络领域的首次应用.

随后, LeCun等人基于Fukushinma的研究工作并使用BP算法设计并训练了LeNet-5, 并在手写数字识别等模式识别领域取得较大的成功. LeNet-5是一经典的CNN(Convolutional Neural Network, 卷积神经网络)模型, 后续很多工作便是基于此进行改进, 例如前先介绍的AlexNet便是改进LeNet-5一举夺得2012年ImageNet竞赛第一名.

较传统MLP神经网络, CNN具有局部连接、权值共享、池化操作及多层结构等优良特性, 其前馈过程更类似于真实生物视觉过程. CNN通过权值共享减少了需要训练的权值个数、降低了网络的计算复杂度, 同时通过池化操作使得网络对输入的局部变换具有一定的不变性如平移不变性、缩放不变性等, 提升了网络的泛化能力. CNN将原始数据直接输入到网络中, 然后隐性地从训练数据中进行网络学习, 避免了手工提取特征、从而导致误差累积的缺点, 其整个分类过程是自动的.

CNN的基本结构由输入层、卷积层(Convolut-ional layer)、池化层(Pooling layer)、全连接层及输出层构成. 卷积层和池化层一般会取若干个, 采用卷积层和池化层交替设置. 整体网络结构如下所示

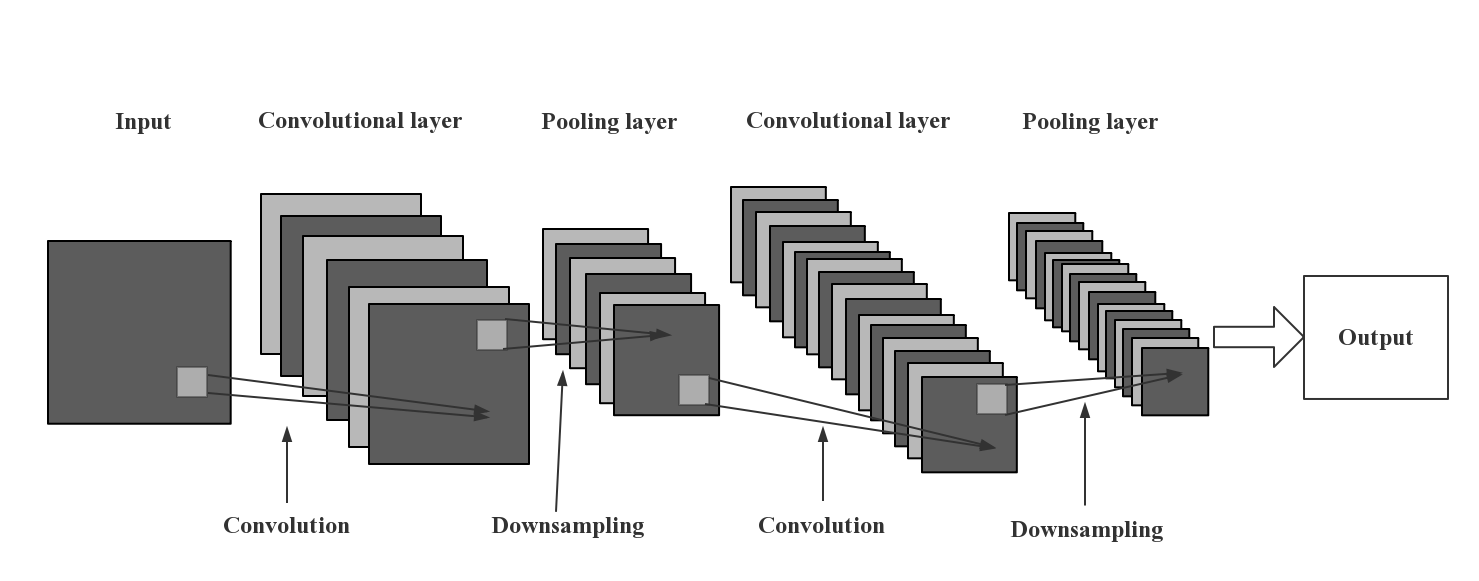


图17 卷积神经网络

**Fig.**17 **Convolutional neural network**

1. **卷积层**

卷积层由多个特征图(Feature Map)组成, 每个特征面由多个神经元组成, 每一个神经元通过卷积核与上一层特征面的局部区域相连. 卷积核是一个权值矩阵, 其作用为提取特征, 第一层提取如边缘、线条、角落等初级特征, 后几层可提取更高级的特征. 卷积阶段的输入为个大小的二维特征图构成的三维数组. 每个特征图记为. 该阶段的输出也是个三维数组, 由个大小的特征图构成. 在卷积阶段, 连接输入特征图和输出特征图的权值记为, 即可训练的卷积核, 卷积核的大小为. 输出特征图为

其中, 为二维离散卷积运算, 示意图如下所示

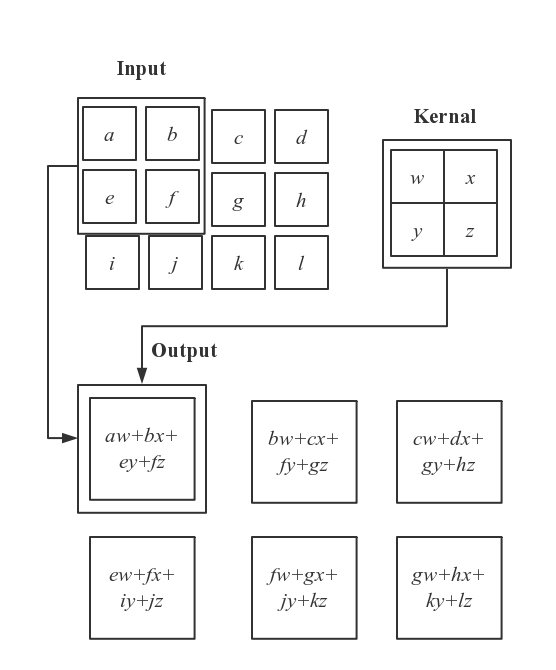


图18 卷积运算

**Fig.**18 **Convolution operation**

对卷积阶段得到的特征作非线性变换, 即“激活”操作, 常用激活函数有

sigmoid

tanh

softsign

ReLU

1. **池化层**

池化层即对输入的特征图作下采样操作, 通常采用平均池化(Average pooling)或最大池化(Max pooling)操作. 平均池化依据定义的邻域窗口计算特定范围内像素的均值, 邻域窗口平移步长大于1(小于等于池化窗口的大小); 最大池化则将均值替换为最值输出到下个阶段. 该层主要目的为降低模型复杂度, 减少运算时间.

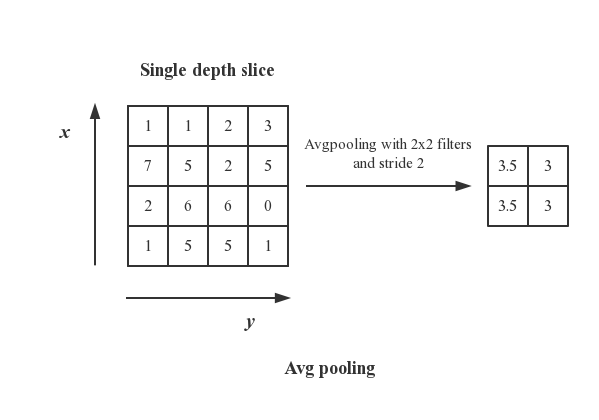


图19 平均池化

**Fig.**19 **Average pooling**

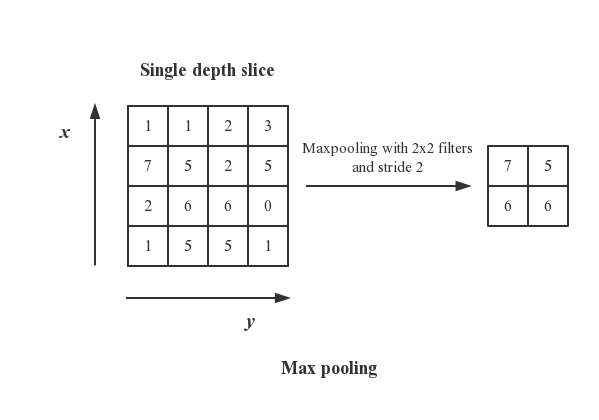


图20 最大池化

**Fig.**20 **Max pooling**

1. **全连接层**

在CNN结构中, 经多个卷积层和池化层后, 连接着1个或1个以上的全连接层. 为了提升CNN网络性能, 全连接层每个神经元的激励函数一般采用ReLU函数. 最后一层全连接层的输出值被传递给一个输出层, 可以采用softmax逻辑回归(softmax regression)进行分类. 通常, CNN的全连接层与MLP的结构一样, CNN的训练算法也多采用BP算法.

# **总结**

目前, 机器学习的研究及应用主要集中在计算机视觉与自然语言处理等领域, 在大数据的驱动下取得了一系列突破性的进展. 例如2016年AlphaGo Lee成功击败了人类棋手李世乭; 2017年, 升级版的AlphaGo Master又击败了人类顶尖棋手柯洁. 此外, 无人驾驶、智能家居等应用也在逐渐成为人们日常生活的主流. 但是机器学习领域仍有许多问题需要进一步研究：

1. 无标记数据的特征学习. 目前标记数据的特征学习仍占主导地位, 而现实生活中存在大量的无标记数据, 逐一采用人工标记是不现实的. 因此随着数据集和存储技术的发展, 无标记数据的特征学习必将成为主流, 未来将越来越重视无标记数据的特征学习, 以及自添加标记算法的研究.
2. 模型规模与训练速度、训练精度之间的权衡. 一般地, 相同数据集下, 模型规模越大, 训练精度越高, 训练速度会越慢. 深度学习模型在以往训练难度相当大, 由于硬件的迅速提升而成为目前机器学习领域的主流, 但目前成熟的模型在保证精度的前提下, 仍需要训练5至7天. 虽然离线训练并不影响训练之后模型的应用, 但是对于模型优化, 诸如模型规模调整、超参数设置、训练时调试等问题, 训练时间会严重影响其效率. 因此, 如何在保证一定的训练精度的前提下, 提高训练速度, 依然是深度学习方向研究的课题之一.
3. 尽管目前机器学习取得了诸多突破, 但仍与真正的“人工智能”之间存在很大差距, 即目前的机器学习算法并不“智能”, 其本质是运用优化算法减少训练时模型的误差, 与真实人类学习过程差距较大. 这一部分的研究不仅依托着对传统算法的改进与突破, 更大程度上需要认知科学领域在人类认知及学习方向取得进一步突破性的成果.

参 考 文 献

1. 周志华.机器学习[M].北京:清华大学出版社,2016.
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning[M]. Massachusetts:The MIT Press,2016
3. 孙志远,鲁成祥,史忠植,马刚.深度学习研究与进展[J].计算机科学,2016,43(02):1-8.
4. 陈凯,朱钰.机器学习及其相关算法综述[J].统计与信息论坛,2007(05):105-112.
5. 马世龙,乌尼日其其格,李小平.大数据与深度学习综述[J].智能系统学报,2016,11(06):728-742.
6. 孙志军,薛磊,许阳明,王正.深度学习研究综述[J].计算机应用研究,2012,29(08):2806-2810.
7. 张润,王永滨.机器学习及其算法和发展研究[J].中国传媒大学学报(自然科学版),2016,23(02):10-18+24.
8. 庄福振,罗平,何清,史忠植.迁移学习研究进展[J].软件学报,2015,26(01):26-39.
9. 刘全,翟建伟,章宗长,钟珊,周倩,章鹏,徐进.深度强化学习综述[J].计算机学报,2018,41(01):1-27.
10. 栾丽华,吉根林.决策树分类技术研究[J].计算机工程,2004(09):94-96+105.
11. Xindong Wu,Vipin Kumar,J. Ross Quinlan,Joydeep Ghosh,Qiang Yang,Hiroshi Motoda,Geoffrey J. McLachlan,Angus Ng,Bing Liu,Philip S. Yu,Zhi-Hua Zhou,Michael Steinbach,David J. Hand,Dan Steinberg. Top 10 algorithms in data mining[J]. Knowledge and Information Systems,2009,14 (1).
12. 李彦冬,郝宗波,雷航.卷积神经网络研究综述[J].计算机应用,2016,36(09):2508-2515+2565.
13. 尹宝才,王文通,王立春.深度学习研究综述[J].北京工业大学学报,2015,41(01):48-59.