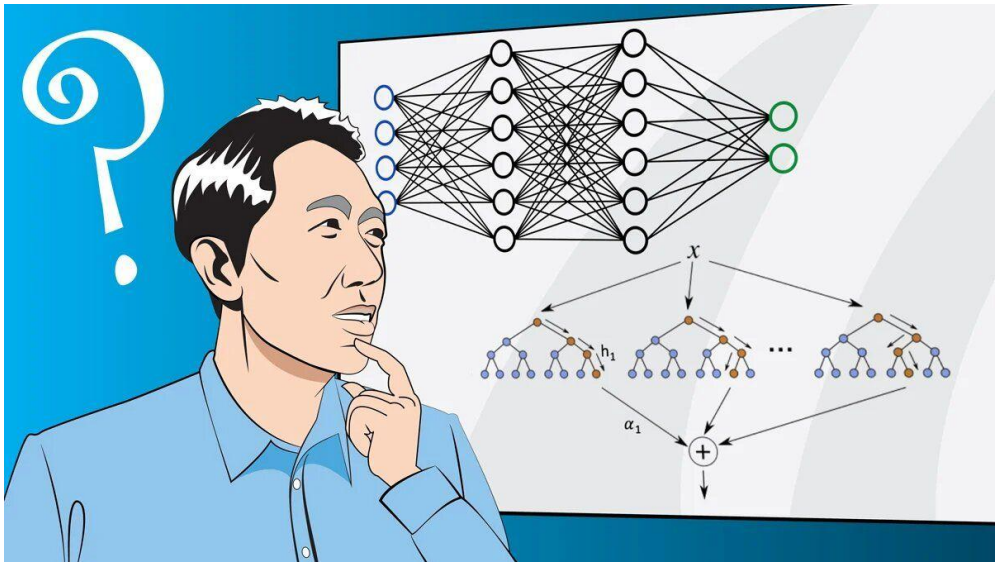
吴恩达：机器学习 6 大核心算法

亲爱的朋友们：

几年前，我不得不在神经网络和决策树学习算法之间做出选择。有必要选择一个高效的算法，因为我们计划在有限的计算预算下将该算法应用于大量用户。我选择了神经网络。我已经有一段时间没有使用增强决策树了，我认为它们需要比实际更多的计算——所以我做了一个错误的决定。幸运的是，我的团队很快修改了我的决定，项目取得了成功。

这次经历让我们明白了学习和不断刷新基础知识的重要性。如果我重新熟悉了提升树，我会做出更好的决定。

与许多技术领域一样，机器学习随着研究人员社区在彼此工作的基础上不断发展而不断发展。有些贡献具有持久力并成为进一步发展的基础。因此，从房价预测器到文本到图像生成器的一切都建立在核心思想之上，包括算法（线性和逻辑回归、决策树等）和概念（正则化、优化损失函数、偏差/方差等）。



坚实、最新的基础是成为一名高效的机器学习工程师的关键。许多团队在日常工作中借鉴这些想法，博客文章和研究论文通常假设您熟悉它们。这种共享的知识基础对于我们近年来机器学习的快速进步至关重要。

本着这种精神，本周的《The Batch》杂志探讨了我们领域中一些最重要的算法，解释了它们的工作原理并描述了它们的一些令人惊讶的起源。如果您刚刚开始，我希望它能够揭开机器学习核心方法的神秘面纱。对于那些更先进的人来说，您会在熟悉的领域发现鲜为人知的观点。不管怎样，我希望这期特刊能够帮助您建立直觉，并为您提供有关机器学习基础的有趣事实，您可以与朋友分享。

保持学习！

吴恩达

模术源论文指导

1 基本算法

机器学习提供了解决各种问题的深层工具箱，但哪种工具最适合哪种任务呢？开口扳手什么时候比活动扳手更好？到底是谁发明了这些东西？在本期《The Batch》特刊中，我们调查了套件中最有用的六种算法：它们来自哪里、做什么，以及随着人工智能深入到社会的各个方面，它们如何演变。如果您想了解更多信息，机器学习专业化提供了对这些算法等的简单实用的介绍。

模术狮论文指导

2 线性回归：直线且窄

线性回归可能是机器学习中的关键统计方法，但它并不是没有经过斗争就成为这样的方法。两位杰出的数学家声称对此有贡献，但 200 年后这个问题仍然没有得到解决。长期存在的争议不仅证明了该算法的非凡实用性，也证明了其本质上的简单性。

到底是谁的算法？

1805 年，法国数学家阿德里安-玛丽·勒让德(Adrien-Marie Legendre)发表了在尝试预测彗星位置时将一条线拟合到一组点的方法（天体导航是当时全球商业中最有价值的科学，就像今天的人工智能一样）——新电力，如果你愿意的话，比电动机早二十年）。

四年后，24 岁的德国神童卡尔·弗里德里希·高斯(Carl Friedrich Gauss)坚称，他自 1795 年以来就一直在使用它，但认为它太微不足道，不值得写下来。高斯的说法促使勒让德匿名发表了一份附录，指出“一位非常著名的几何学家毫不犹豫地采用了这种方法。”

斜率和偏差：只要结果与影响结果的变量之间的关系呈直线，线性回归就很有用。例如，汽车的油耗与其重量呈线性关系。

- 汽车的油耗 y 与其重量 x 之间的关系取决于直线的斜率 w（油耗随重量增加的陡峭程度）和偏差项 b（零重量时的油耗）：y=w*x+b。
- 在训练过程中，根据汽车的重量，算法会预测预期的油耗。它比较预期油耗和实际油耗。然后，它通常通过普通最小二乘技术来最小化平方差，该技术会磨练 w 和 b 的值。
- 考虑到汽车的阻力可以产生更精确的预测。附加变量将线延伸成平面。这样，线性回归可以容纳任意数量的变量/维度。

普及的两个步骤：该算法立即帮助航海家追随星星，并帮助后来的生物学家（特别是查尔斯·达尔文的表弟弗朗西斯·高尔顿）识别动植物的遗传特征。两项进一步的发展释放了其广泛的潜力。1922 年，英国统计学家罗纳德·费舍尔(Ronald Fisher)和卡尔·皮尔逊(Karl Pearson)展示了线性回归如何融入相关性和分布的一般统计框架，从而使其在所

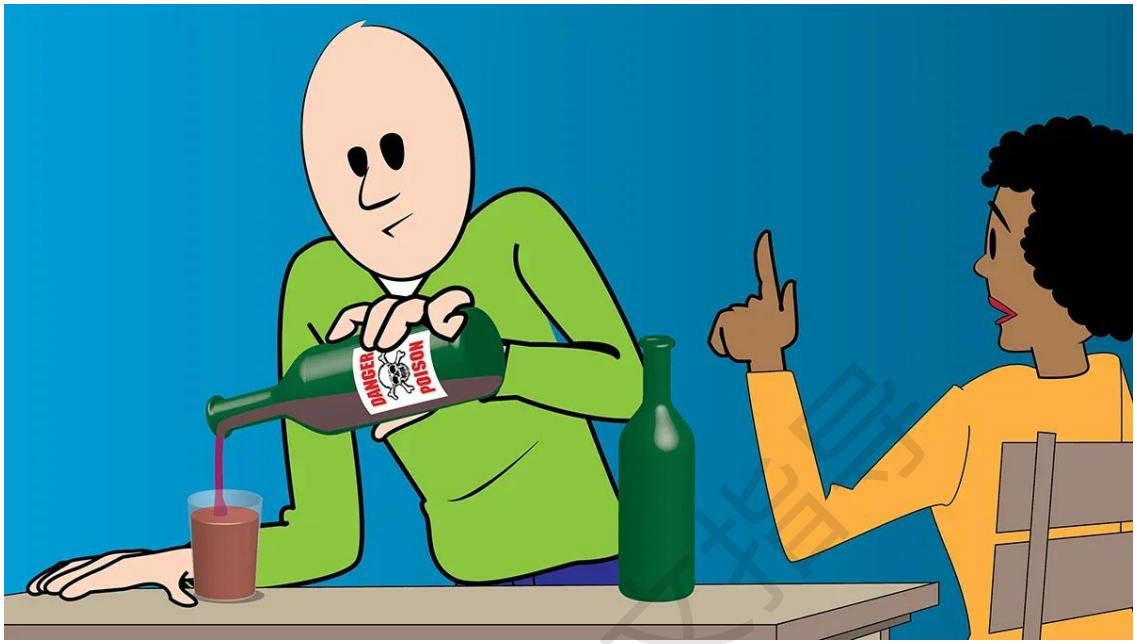
有科学领域中发挥作用。近一个世纪后，计算机的出现提供了数据和处理能力，可以更好地利用它。

应对模糊性：当然，数据永远无法完美测量，并且某些变量比其他变量更重要。这些生活事实催生了更复杂的变体。例如，具有正则化的线性回归（也称为岭回归）鼓励线性回归模型不要过多依赖于任何一个变量，或者更确切地说均匀地依赖于最重要的变量。如果您追求简单性，则可以使用不同形式的正则化（L1 而不是 L2）产生 lasso，这会鼓励尽可能多的系数为零。换句话说，它学会选择具有高预测能力的变量并忽略其余的。弹性网络结合了两种类型的正则化。当数据稀疏或特征似乎相关时，它很有用。

在每个神经元中：尽管如此，简单的版本还是非常有用的。神经网络中最常见的神经元类型是线性回归模型，后跟非线性激活函数，使线性回归成为深度学习的基本构建块。

模板论文指导

3 逻辑回归：跟随曲线



曾经有一段时间，逻辑回归只被用来对一件事进行分类：如果你喝了一瓶毒药，你可能会被贴上“活着”或“死者”的标签吗？时代已经变了，如今，呼叫紧急服务不仅可以为这个问题提供更好的答案，而且逻辑回归也是深度学习的核心。

毒物控制：物流功能可以追溯到 1830 年代，当时比利时统计学家 P.F. Verhulst 发明它是为了描述人口动态：随着时间的推移，最初的指数增长随着消耗可用资源而趋于平缓，从而形成特征逻辑曲线。一个多世纪过去了，美国统计学家 E.B.Wilson 和他的学生 Jane Worcester 设计了逻辑回归来计算出某种危险物质的致命程度。他们如何收集训练数据是另一篇文章的主题。

拟合函数：逻辑回归将逻辑函数拟合到数据集，以便预测给定事件（例如摄入马钱子碱）时发生特定结果（例如过早死亡）的概率。

- 训练会水平调整曲线的中心位置和垂直方向的中间位置，以最大限度地减少函数输出和数据之间的误差。
- 将中心调整到右边或左边，意味着要杀死一般人需要或多或少的毒药。陡峭的斜坡意味着确定性：在中点之前，大多数人都生存下来；过了一半，再见。平缓的斜坡比较宽容：低于曲线的中间，一半以上的人可以生存；再往上，还不到一半。
- 在一个结果与另一个结果之间设置一个阈值，例如 0.5，曲线就成为一个分类器。只需将剂量输入模型，您就会知道是否应该计划聚会或葬礼。

更多结果：Verhulst 的工作发现了二元结果的概率，忽略了进一步的可能性，例如中毒受害者可能落在来世的哪一边。他的继任者扩展了算法：

- 英国统计学家 David Cox 和荷兰统计学家 Henri Theil 在 20 世纪 60 年代末独立工作，将逻辑回归应用于具有两种以上可能结果的情况。
- 进一步的工作产生了有序逻辑回归，其中结果是有序值。
- 为了处理稀疏或高维数据，逻辑回归可以利用与线性回归相同的正则化技术。

多功能曲线：逻辑函数以相当的准确性描述了各种现象，因此逻辑回归在许多情况下提供了有用的基线预测。在医学上，它估计死亡

摸术源论文指导

率和疾病风险。在政治学中，它预测选举的赢家和输家。在经济学中，它预测商业前景。更重要的是，它驱动多种神经网络中的一部分神经元，其中非线性为 sigmoid。

模术狮论文指导

4 梯度下降：都是下坡路

想象一下黄昏后在山中徒步旅行，发现脚下看不到太多东西。而且您的手机电池没电了，因此您无法使用 GPS 应用程序找到回家的路。您可能会通过梯度下降找到最快的路径。只是要小心不要走下悬崖。

太阳和地毯：梯度下降不仅仅适合在陡峭的地形中下降。1847 年，法国数学家奥古斯丁·路易斯·柯西发明了近似恒星轨道的算法。六十年后，他的同胞雅克·阿达玛（Jacques Hadamard）独立开发了它来描述薄而灵活的物体（例如小地毯）的变形，这可能会使膝盖更容易向下徒步。然而，在机器学习中，它最常见的用途是找到学习算法损失函数的最低点。

向下爬：经过训练的神经网络提供一个函数，根据给定的输入，计算所需的输出。训练网络的一种方法是通过迭代计算实际输出与期望输出之间的差异，然后更改网络的参数值以缩小该差异，从而最大

限度地减少其输出中的损失或错误。梯度下降缩小了差异，最小化了计算损失的函数。网络的参数值相当于景观上的一个位置，损失是当前的高度。当你下降时，你会提高网络计算接近所需输出的能力。可见性是有限的，因为在典型的监督学习情况下，算法仅依赖于网络的参数值和损失函数的梯度或斜率，即您在山上的位置以及您脚下的斜坡。

- 基本方法是向地形下降最陡的方向移动。诀窍是校准你的步幅。太小了，需要很长时间才能取得进展。太大了，你就会跳入未知的世界，可能会上坡而不是下坡。
- 给定当前位置，该算法通过计算损失函数的梯度来估计最速下降的方向。梯度指向上坡，因此算法通过减去梯度的一小部分来向相反的方向前进。分数α称为学习率，决定再次测量梯度之前的步长大小。
- 迭代地应用这个，希望你能到达一个山谷。恭喜！

卡在山谷里：太糟糕了，你的手机没电了，因为算法可能无法将你推到凸山的底部。您可能会陷入由多个山谷（局部最小值）、峰（局部最大值）、鞍点（鞍点）和高原组成的非凸景观中。事实上，图像识别、文本生成和语音识别等任务都是非凸的，并且已经出现了梯度下降的许多变体来处理这种情况。例如，该算法可能具有动量，可以帮助其放大小幅上升和下降，从而更好地到达底部。研究人员设计了如此多的变体，看起来好像有多少个局部最小值就有多少个优化器。

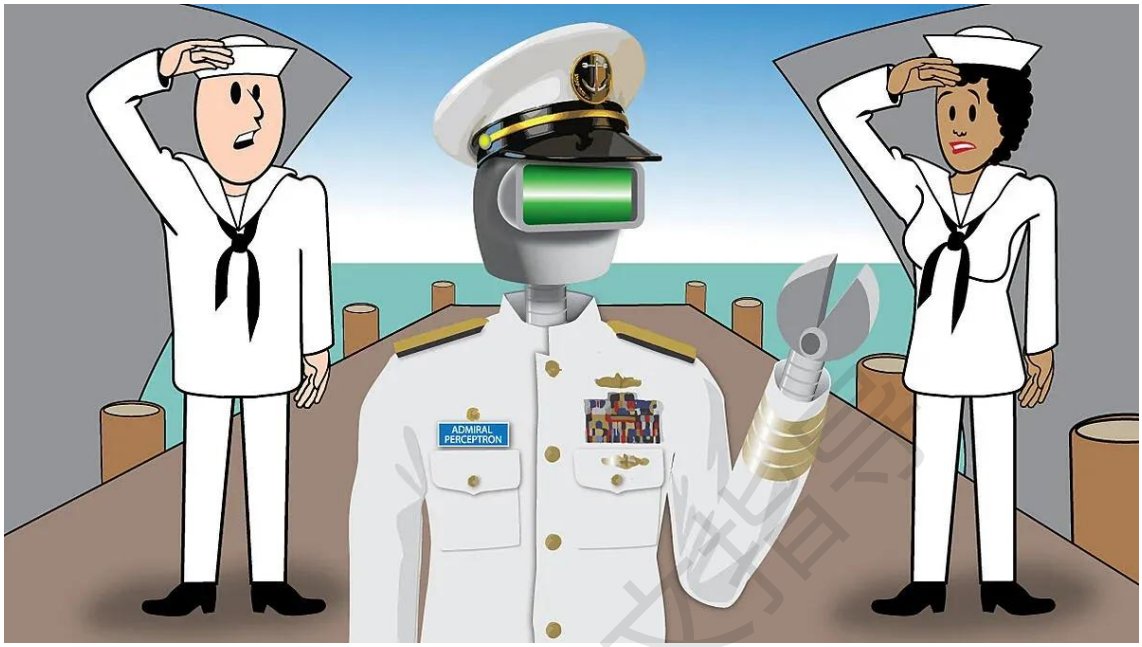
模术源论文指导

幸运的是，局部和全局最小值往往大致相同。

最佳优化器：梯度下降是寻找任何函数最小值的明智选择。在可以直接计算精确解的情况下（例如，具有大量变量的线性回归任务），它可以逼近一个解，通常更快、更便宜。但它在复杂的非线性任务中确实发挥了作用。凭借梯度下降和冒险精神，您可能会在晚餐时间及时出山。

模术狮论文指导

5 神经网络：求函数



让我们把这个问题弄清楚：大脑不是图形处理单元的集群，如果是的话，它会运行比典型的人工神经网络复杂得多的软件。然而，神经网络受到了大脑结构的启发：互连的神经元层，每个神经元根据其邻居的状态计算自己的输出。由此产生的级联活动形成了一个想法——或者识别出了一张猫的图片。

从生物到人工：大脑通过神经元之间的相互作用进行学习的见解可以追溯到 1873 年，但直到 1943 年，美国神经科学家 Warren McCulloch 和 Walter Pitts 才使用简单的数学规则对生物神经网络进行建模。1958 年，美国心理学家 Frank Rosenblatt 开发了感知器，这是一种在打孔卡上实现的单层视觉网络，旨在为美国海军构建硬件版本。

越大越好：罗森布拉特的发明只识别可以用线分隔的类。乌克兰数学家 Alexey Ivakhnenko 和 Valentin Lapa 通过堆叠任意层数的神经元

网络克服了这一限制。1985 年，法国计算机科学家 Yann LeCun、David Parker 和美国心理学家 David Rumelhart 及其同事独立工作，描述了使用反向传播来有效地训练此类网络。在新千年的第一个十年，包括 Kumar Chellapilla、Dave Steinkraus 和 Rajat Raina（与 Andrew Ng）在内的研究人员使用图形处理单元加速了神经网络，这使得越来越大的神经网络能够从生成的大量数据中学习通过互联网。

适合每项任务：神经网络背后的想法很简单：对于任何任务，都有一个可以执行它的函数。神经网络通过组合许多简单的函数来构成可训练的函数，每个函数都由单个神经元执行。神经元的功能由称为权重的可调参数决定。给定这些权重的随机值以及输入及其所需输出的示例，可以迭代地改变权重，直到可训练函数执行手头的任务。

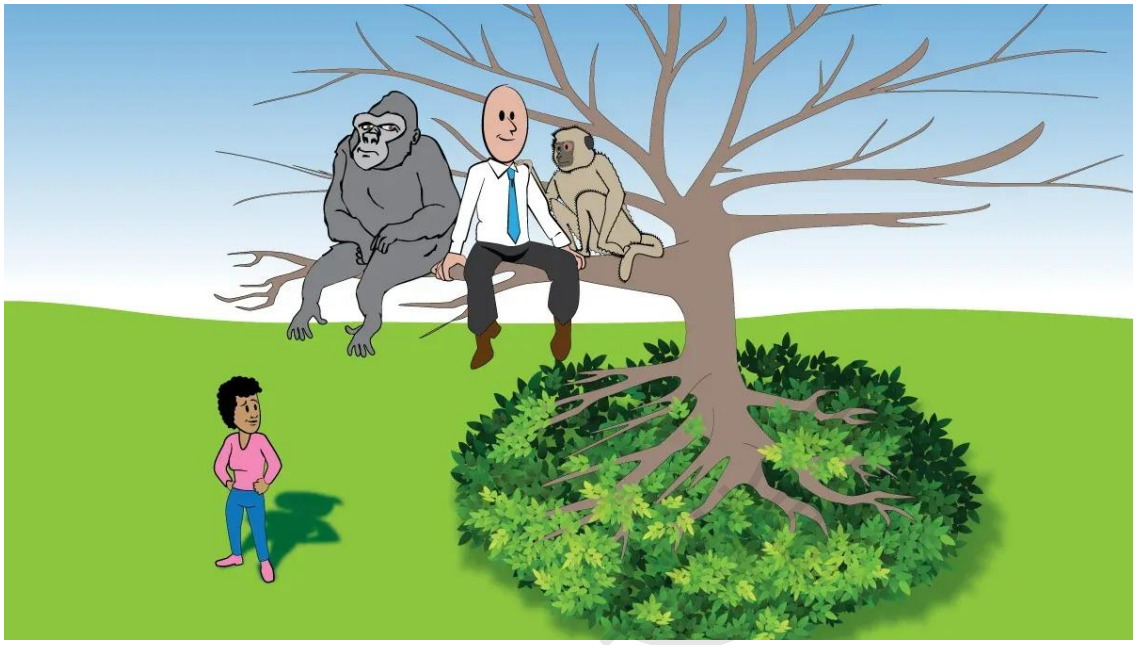
- 神经元接受各种输入（例如，代表像素或单词的数字，或前一层的输出），将它们与其权重相乘，将乘积相加，然后通过非线性函数或激活函数提供总和，该非线性函数或激活函数由下式选择：开发商。考虑它是线性回归加上激活函数。
- 训练会修改权重。对于每个示例输入，网络都会计算输出并将其与预期输出进行比较。反向传播使用梯度下降来改变权重，以减少实际输出和预期输出之间的差异。使用足够（好的）示例重复此过程足够多次，网络应该学会执行该任务。

黑盒：虽然经过训练的网络可以执行其任务，但祝你好运确定如何执行。您可以阅读最终的函数，但它通常非常复杂-具有数千个变量

和嵌套的激活函数-以至于很难解释网络如何成功完成其任务。此外，经过训练的网络的好坏取决于它所学习的数据。例如，如果数据集有偏差，网络的输出也会有偏差。如果它只包含猫的高分辨率图片，就无法知道它会对低分辨率图像有何反应。

一个常识：1958 年，《纽约时报》在报道罗森布拉特的感知器时，为人工智能的炒作开辟了道路，称其为“美国海军期望能够行走、说话、看、写、再现的电子计算机的胚胎”并意识到它的存在。”虽然它没有达到这个标准，但它产生了许多令人印象深刻的模型：图像的卷积神经网络；用于文本的循环神经网络；以及图像、文本、语音、视频、蛋白质结构等的转换器。他们做了令人惊奇的事情，在下围棋方面超越了人类水平，并在诊断 X 射线图像等实际任务中接近人类水平。然而，他们仍然很难掌握常识和逻辑推理。

6 决策树：从根到叶



亚里士多德是什么样的野兽？这位哲学家的追随者波菲利生活在三世纪的叙利亚，他想出了一个合乎逻辑的方法来回答这个问题。他将亚里士多德提出的“存在范畴”从一般到具体进行了整理，并依次将亚里士多德本人归入每个范畴：亚里士多德的实体占据空间，而不是概念或精神；他的身体是有生命的，而不是无生命的；他的头脑是理性的，而不是非理性的。因此他的分类是人类的。中世纪的逻辑教师将这个序列绘制为垂直流程图：早期的决策树。

数字差异：快进到 1963 年，当时密歇根大学社会学家 John Sonquist 和经济学家 James Morgan 将调查受访者分组，首先在计算机中实现了决策树。随着自动训练算法的软件的出现，此类工作变得司空见惯，目前已在包括 scikit-learn 在内的各种机器学习库中实现。斯坦福大学和加州大学伯克利分校的四位统计学家花了 10 年时间才开发出该代码。

如今，从头开始编写决策树是机器学习 101 中的一项家庭作业。

根在天空：决策树可以执行分类或回归。它在决策层次结构中从根部到树冠向下生长，将输入示例分为两个（或更多）组。想想约翰·布卢门巴赫(Johann Blumenbach)的任务，他是一位德国医生和人类学家，大约在 1776 年首次将猴子与猿类（人类除外）区分开来，在此之前，它们被归为一类。分类取决于各种标准，例如是否有尾巴、胸部狭窄或宽阔、直立姿势还是蹲伏姿势以及智力高低。经过训练来标记此类动物的决策树将一一考虑每个标准，最终将两组动物分开。

- 该树从根节点开始，可以将其视为包含生物数据集中的所有示例-黑猩猩、大猩猩和猩猩以及卷尾猴、狒狒和狨猴。根代表在表现出特定特征或不表现出特定特征的示例之间进行选择，从而导致两个子节点包含具有和不具有该特征的示例。每个孩子都会提出另一个选择（有或没有不同的特征），导致另外两个孩子，依此类推。该过程以任意数量的叶节点结束，每个叶节点都包含大部分或全部属于一个类的示例。
- 为了成长，树必须找到根决策。在选择时，它会考虑所有特征及其值（后附肢、桶状胸等），并选择能够最大化分割纯度的特征，最佳纯度定义为某一类别的示例 100%属于特定儿童节点，并且没有一个节点到达另一个节点。仅仅做出一个决定后，拆分很少是 100%纯净的，并且可能永远不会达到这一目标，因此该过程会继续，产生一层又一层的子节点，直到通过考虑进一步的功能，纯度不会提高太多。至此，树已经完全训练完毕。

模不得论文指导

- 在推理时，一个新的例子从上到下遍历树，评估每个级别的不同决策。它采用它所在的叶节点包含的数据的标签。

进入前 10 名：考虑到布卢门巴赫的结论（后来被查尔斯·达尔文推翻），即人类与猿类的区别在于宽阔的骨盆、手和紧密排列的牙齿，如果我们想扩展决策树，不仅可以对猿类和猴子进行分类，还可以对猿类和猴子进行分类，该怎么办？人类也一样吗？澳大利亚计算机科学家 John Ross Quinlan 于 1986 年通过 ID3 使这成为可能，它扩展了决策树以支持非二元结果。2008 年，IEEE 国际数据挖掘会议策划的数据挖掘十大算法列表中，名为 C4.5 的进一步改进被列入其中。在创新猖獗的世界里，这就是持久力。

扒开树叶：决策树确实有一些缺点。他们可以通过增长如此多的级别来轻松地过度拟合数据，以至于叶节点只包含一个示例。更糟糕的是，它们很容易产生蝴蝶效应：改变一个例子，长出的树可能看起来会截然不同。

走进森林：将这一特征转化为优势，美国统计学家 Leo Breiman 和新西兰统计学家 Adele Cutler 在 2001 年开发了随机森林，这是一个决策树集合，每个决策树都会处理不同的、重叠的示例选择，并对最终结果进行投票决定。随机森林及其表兄弟 XGBoost 不太容易出现过度拟合，这有助于使它们成为最流行的机器学习算法之一。这就像亚里士多德、波菲利、布卢门巴赫、达尔文、简、戴安·福西和其他 1,000 名动物学家在一个房间里，所有人都确保你的分类是最好的。

模不通论文指导

7 K 均值聚类：群体思考

如果您在聚会上与其他人站得很近，那么你们很可能有一些共同点。这就是使用 k 均值聚类将数据点分成组的想法。无论这些团体是通过人类机构还是其他力量形成的，该算法都会找到它们。

从爆炸到拨号音：美国物理学家斯图尔特·劳埃德（Stuart Lloyd）是贝尔实验室标志性创新工厂和发明原子弹的曼哈顿计划的校友，他于 1957 年首次提出 k 均值聚类来在数字信号中分配信息。直到 1982 年他才发表该算法：

Least Squares Quantization in PCM

STUART P. LLOYD

论文地址：https://cs.nyu.edu/~roweis/csc2515-2006/readings/lloyd57.pdf

与此同时，美国统计学家 Edward Forgy 在 1965 年描述了一种类似的方法，并因此得名 Lloyd-Forgy 算法。

找到中心：考虑将党分成志同道合的工作组。给定与会者在房间中的位置和要形成的组的数量，k 均值聚类可以将与会者分成大小大致相等的组，每个组聚集在一个中心点或质心周围。

- 在训练过程中，算法最初通过随机选择 k 个人来指定 k 个质心。（K 必须手动选择，并且找到最佳值并不总是微不足道的。）然后，它通过将每个人与最近的质心相关联来增长 k 个簇。
- 对于每个集群，它计算分配到该组的所有人员的平均位置，并将平均位置指定为新的质心。每个新的质心可能都没有被人占据，但那又怎样呢？人们倾向于聚集在巧克力火锅周围。
- 对于每个集群，它计算分配到该组的所有人员的平均位置，并将平均位置指定为新的质心。每个新的质心可能都没有被人占据，但那又怎样呢？人们倾向于聚集在巧克力火锅周围。
- 预先警告：考虑到最初的随机质心分配，您最终可能不会与您希望的那个可爱的以数据为中心的人工智能专家在同一组中。该算法做得很好，但不能保证找到最佳解决方案。祝下次聚会好运！

不同的距离：当然，聚类对象之间的距离不需要是空间距离。两个向量之间的任何测量都可以。例如，k 均值聚类可以根据服装、职业或其他属性对参加派对的人进行划分，而不是根据物理距离对其进行分组。在线商店使用它根据顾客的喜好或行为来划分顾客，天文学家使用它来对相同类型的恒星进行分组。

数据点的力量：这个想法产生了一些显着的变化：

- K 中心点使用实际数据点作为质心，而不是给定簇中的平均位置。中心点是与簇中所有其他点的距离最小化的点。这种变化更容易解释，因为质心始终是数据点。
- 模糊 C 均值聚类使数据点能够不同程度地参与多个聚类。它根据距质心的距离，用隶属程度取代了硬聚类分配。

n 维狂欢：尽管如此，原始形式的算法仍然广泛有用，特别是因为，作为一种无监督算法，它不需要收集可能昂贵的标记数据。使用起来也更快。例如，包括 scikit-learn 在内的机器学习库受益于 2002 年添加的 kd 树，它可以极快地划分高维数据。顺便说一句，如果您举办任何高规格的派对，我们很乐意出现在宾客名单上。

模术狮论文指导