

1.1 极限的概念与定义

极限是微积分的基础，其基本概念与定义具有至关重要的作用。一般来说，极限主要描述了一个数列或函数在某一点或无穷远处的趋向行为。具体定义如下。

数列极限：对于数列 $\{a_n\}$ ，如果存在实数 L ，对于任意给定的正数 ε （无论多么小），总存在正整数 N ，使得当 $n > N$ 时， $|a_n - L| < \varepsilon$ ，称 L 是数列 $\{a_n\}$ 的极限，记作 $\lim_{n \rightarrow \infty} a_n = L$ 。

函数极限：设函数 $f(x)$ 在点 x_0 的某一去心邻域内有定义， A 为常数，如果对于任意给定的正数 ε ，总存在 $\delta(>0)$ ，使得当 $0 < |x - x_0| < \delta$ 时， $|f(x) - A| < \varepsilon$ ，那么称 A 为函数 $f(x)$ 在 x_0 处的极限，记作 $\lim_{x \rightarrow x_0} f(x) = A$ 。

无穷小量：如果一个数列或函数的极限等于 0，那么我们称这个数列或函数为无穷小量。

无穷大量：如果一个数列或函数的绝对值的极限等于无穷大，那么我们称这个数列或函数为无穷大量。

极限的存在性是需要证明的，而且并非所有的数列或函数都有极限。此外，极限存在不代表函数在极限点处有定义。极限的概念帮助我们理解无限的概念，并为微积分中的许多其他重要概念提供了基础，例如连续性、微分和积分。

在接下来的章节中，我们将详细讨论极限的性质以及极限与连续的关系，以深化读者对这一重要概念的理解。



无穷小量的性质



无穷小量与无穷大量的关系

1.2 极限的性质

极限的性质是处理极限问题的重要工具，也是微积分理论的基石。在了解极限的定义之后，我们可以研究极限的一些基本性质，这些性质将为求解具体的极限问题时提供帮助。以下是极限的一些基本性质。

唯一性：如果数列或函数在某一点的极限存在，那么这个极限是唯一的。换言之，不可能有两个不同的数都是同一个数列或函数的极限。

有界性：如果数列或函数在某一点有极限，那么在这一点的某个邻域内，数列或函数必定是有界的。

保序性：如果两个数列或函数在某一点的极限存在，并且在这一点的某个邻域内数列或函数保持某种序关系，那么这种序关系在极限处依然保持。

算术运算性质：极限可加、可减、可乘、可除。也就是说，如果两个数列或函数都在某一点有极限，那么它们的和、差、积、商（除数不为 0）在这一点的极限等于这两个极限的和、差、积、商。

夹逼定理：如果函数 $f(x)$ 、 $g(x)$ 、 $h(x)$ 满足 $g(x) \leq f(x) \leq h(x)$ 且 $\lim_{x \rightarrow x_0} g(x) = \lim_{x \rightarrow x_0} h(x) = a$ ，那么

$$\lim_{x \rightarrow x_0} f(x) = a。$$

以上只是极限的一些基本性质，实际上极限的性质还有很多，例如复合函数的极限性质、极限与无穷小量的关系等。了解这些性质能够帮助我们更好地理解和应用极限这一概念。

1.3 连续的概念与定义

在微积分中，连续性是一个重要的基本概念，它是描述函数在某点附近行为的方式。下面给出函数在某一点处连续的定义。

假设函数 $f(x)$ 定义在某区间 I 上，如果对于 I 上的一个点 c ， $x=c$ 处的极限值等于函数在该点的函数值 $f(c)$ ，即

$$\lim_{x \rightarrow c} f(x) = f(c)$$

则称函数 $f(x)$ 在点 c 处连续。换句话说，函数在某点连续意味着函数在这一点的行为是可预测的，没有跳跃或突变。

更进一步，如果函数 f 在区间 I 的所有点上连续，那么就说函数 f 在 I 上连续。

另一种直观的理解是，如果函数图像是一条不断的曲线，那么这个函数就是连续的。

这个概念在微积分中至关重要，因为许多微积分的基本理论（如中值定理、微分定理、积分定理等）都需要函数的连续性这个前提条件。

1.4 连续的性质

连续函数具有以下一些重要的性质，这些性质对连续函数的分析和处理非常有用。

介值定理：如果函数 f 在闭区间 $[a, b]$ 上连续，且对于某些 c ， $f(a) < c < f(b)$ ，那么在区间 (a, b) 内存在某点 x ，使得 $f(x) = c$ 。换句话说，连续函数能够取到其最大值和最小值之间的任何值。

最值定理：如果函数 f 在闭区间 $[a, b]$ 上连续，那么 f 在 $[a, b]$ 上必然可以取到最大值和最小值。这个定理说明，任何在闭区间上连续的函数其图像都是“封闭”的，不会“逃逸”到无穷大或无穷小。

运算法则：连续函数的和、差、积、商（除数不为 0）都是连续的。这意味着如果有两个连续函数，可以将它们相加、相减、相乘、相除（除数不为 0），其结果仍然是连续的。

复合函数的连续性：如果函数 g 在点 a 处连续，函数 f 在点 $g(a)$ 处连续，那么复合函数 $f(g(x))$ 在点 a 处连续。

反函数的连续性：如果函数 f 在闭区间 I 上单调连续，那么它的反函数在其值域上连续。这些性质在高等数学中有着广泛的应用，特别是在微分学、积分学和微分方程等领域。

1.5 极限与连续的关系

极限与连续的概念是微积分的基础，并且它们之间存在密切的关系。这种关系在定义连续函数时就体现出来了。

定义连续：一个函数 $f(x)$ 在点 $x = x_0$ 上连续，如果满足以下条件：

- (1) 函数 $f(x)$ 在点 $x = x_0$ 处有定义；
- (2) 函数在点 $x = x_0$ 处的左极限和右极限都存在且相等；
- (3) 函数在点 $x = x_0$ 处的值等于该点的极限。

换句话说，一个函数在某一点连续，也就是当自变量无限接近这一点时，函数的值也无限接近于该点的函数值。这就是极限与连续性之间的基本关系。

此外，极限与连续还在以下几个重要的定理中展示了它们的关系。

介值定理: 如果函数在闭区间 $[a, b]$ 上连续, 并且 $f(a)$ 不等于 $f(b)$, 那么对于任意介于 $f(a)$ 和 $f(b)$ 之间的数都有一个 c 属于 $[a, b]$, 使得 $f(c)$ 等于该数。这个定理的证明依赖于连续函数的极限性质。

最值定理: 一个在闭区间 $[a, b]$ 上连续的函数必定在该区间上有最大值和最小值。这个定理的证明也利用了极限的性质。

零点定理: 如果闭区间 $[a, b]$ 上的连续函数 f 在两个端点上的函数值异号, 即 $f(a)f(b) < 0$, 那么开区间 (a, b) 内至少存在一点 ξ , 使得 $f(\xi) = 0$ 。这个定理的证明依赖于连续函数的介值定理, 而介值定理又与极限的概念密切相关。

在高等数学的许多领域中, 极限与连续都起着重要的作用。理解它们的关系可以帮助我们更好地理解并应用这些概念。

1.6 铁人三项中的最佳体能分配简化模型和数值计算实践

铁人三项是一种极限体能挑战, 由游泳、自行车骑行和跑步 3 个连续项目组成。对于参加者来说, 最佳的体能分配对于取得优异成绩至关重要。如何使用极限的概念来解决这个问题呢?

首先, 可以将铁人三项的每个项目都视为一个变量, 我们要找的就是这 3 个变量的最佳组合, 以达到整体完成时间的极限 (也就是最小值)。为了找到这个组合, 我们可以借助微积分中的极值理论。

具体来说, 我们可以设定一个函数, 该函数的输入是针对 3 个项目的体能分配, 输出是预期的完成时间。这个函数会在某一体能分配下达到极小值。这个极小值的位置就是最佳的体能分配。

然而, 这个函数非常复杂, 涉及许多因素, 比如参赛者的体能、天气、路线等。要找到这个函数的极小值, 需要借助高级的优化算法, 比如梯度下降法。

更具体的方法可能是通过训练和比赛的数据建立一个模型来预测不同体能分配下的完成时间, 然后使用优化算法找到使完成时间最短的体能分配。这样的优化问题在实际生活中经常出现, 解决这类问题的方法也是高等数学的一个重要应用。

通过这个案例, 我们可以看到极限概念在实际问题中的应用, 也可以看到高等数学在体育等领域的应用。

上述的数学表达

我们可以使用微积分中的最优化理论。设 $F(x, y, z)$ 是完成铁人三项的总时间, x 、 y 、 z 分别代表在游泳、自行车骑行和跑步上分配的体能比例。

我们的目标是找到一组体能分配组合 (x, y, z) , 使得 $F(x, y, z)$ 达到最小。形式上, 这是一个优化问题, 可以用拉格朗日乘数法来解决, 因为体能分配有一个限制条件: $x + y + z = 1$ 。

拉格朗日函数为 $L(x, y, z, \lambda) = F(x, y, z) + \lambda(1 - x - y - z)$ 。我们要找到使得 L 取得极小值的 x 、 y 、 z 和 λ 。

为了找到最小值, 我们需要找到满足以下条件的点 (x, y, z, λ) : L 对 x 、 y 、 z 和 λ 的偏导数等于 0。形式上, 这组条件可以表示为

$$\begin{cases} \frac{\partial L}{\partial x} = \frac{\partial F}{\partial x} - \lambda = 0 \\ \frac{\partial L}{\partial y} = \frac{\partial F}{\partial y} - \lambda = 0 \\ \frac{\partial L}{\partial z} = \frac{\partial F}{\partial z} - \lambda = 0 \\ \frac{\partial L}{\partial \lambda} = 1 - x - y - z = 0 \end{cases}$$

在满足这 4 个条件的点处, L 有可能取得极小值。

这是一个一阶线性微分方程，我们可以用数学的方法来解出。

然后，我们需要找到满足特定条件（例如，炸弹落在目标位置）的解，这就需要我们解决一个极值问题。例如，我们可能需要找到一个角度 θ ，在这个角度下投掷炸弹，炸弹的落点距离目标位置的距离最小。这可以通过解决下面的优化问题来实现：

$$\min f(\theta) = \|\text{投掷炸弹后的落点} - \text{目标点}\|$$

其中， f 是目标函数，我们需要找到最小化这个函数的 θ 。这个问题可以通过优化算法来解决，例如梯度下降法。

最后，通过这种方式，我们可以找到一个最优的角度，使得炸弹能够准确地落在目标位置。这个过程显示了微分方程和优化问题在实际问题中的重要应用。

数值计算实践

实际编程环境中，模拟这样的问题可以涉及使用数值解法求解微分方程，比如欧拉方法或者 Runge-Kutta 方法。这里我们简化问题，假设空气阻力与速度成正比，而且仅在 2D 平面进行投掷。

扫描下方二维码可查看代码。在这个 Python 示例代码中，通过微分方程来模拟炸弹的飞行轨迹，并绘制出飞行路径。使用 NumPy 和 Matplotlib 库定义了物理常数和微分方程，其中微分方程描述了炸弹在空中的运动情况，考虑了重力和空气阻力。设置了初始条件，包括位置和速度，并生成时间向量。通过调用 odeint 函数求解微分方程，得到炸弹在不同时间点的位置和速度信息。最后，使用 Matplotlib 绘制飞行轨迹图，其中 x 轴表示水平位置， y 轴表示垂直位置，如图 1.2 所示。

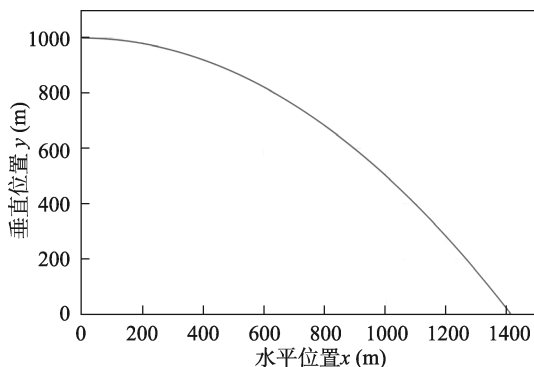


图 1.2 炸弹轨迹



示例代码

1.8 攀登山峰时的营地选择简化模型和数值计算实践

在登山过程中，如何选择营地是一个十分重要的问题，因为营地位置的选择不仅会影响登山者的安全，还会影响登山的进度和成功率。

我们可以使用高等数学中的极限和连续的概念来帮助我们选择营地。以攀登珠穆朗玛峰为例，首先，我们可以通过观测和测量，获取珠峰各个高度处的气温、风力、降雪量等数据，然后利用这些数据建立模型。这些模型通常会表现为一些函数，用以表示气温、风力、降雪量等与高度的关系。

我们可以利用极限的概念来考虑一些极端情况。例如，如果气温降到了极限低温，或者风力达到了极限大风，那么哪些位置会成为危险的营地。另外，我们也可以利用连续的概念来考虑气温、风力、降雪量等随着高度的连续变化，从而找出可能的安全营地。

这个案例展示了极限与连续在登山策略选择中的应用，也展示了高等数学与实际问题的紧密联系。

上述的数学表达

我们可以用函数 $T(h)$ 、 $W(h)$ 和 $S(h)$ 来表示气温、风力和降雪量随高度 h 的变化。我们的目标

是找到一组高度 h ，使得 $T(h)$ 、 $W(h)$ 和 $S(h)$ 满足安全的条件。这是一个约束优化问题，可以用拉格朗日乘数法来解决。

设 $L(h, \lambda_1, \lambda_2, \lambda_3) = h + \lambda_1[T_0 - T(h)] + \lambda_2[W_0 - W(h)] + \lambda_3[S_0 - S(h)]$ 。其中， T_0 、 W_0 和 S_0 是气温、风力和降雪量的安全阈值， λ_1 、 λ_2 和 λ_3 是拉格朗日乘数。

我们的目标是找到一组 h 和 λ ，使得 L 达到最大。形式上，我们需要找到满足以下条件的点：

$$\begin{cases} \frac{\partial L}{\partial h} = 1 - \lambda_1 \frac{dT(h)}{dh} - \lambda_2 \frac{dW(h)}{dh} - \lambda_3 \frac{dS(h)}{dh} = 0 \\ \frac{\partial L}{\partial \lambda_1} = T_0 - T(h) = 0 \\ \frac{\partial L}{\partial \lambda_2} = W_0 - W(h) = 0 \\ \frac{\partial L}{\partial \lambda_3} = S_0 - S(h) = 0 \end{cases}$$

在满足这 4 个条件的点处， L 函数可能达到最大值，这就是可能的营地位置。

然而，真实的情况可能复杂得多，因为气温、风力和降雪量可能与时间和地点等其他因素有关，而且可能不是连续的。在这种情况下，我们需要借助更复杂的数学模型和算法，例如非线性优化或者模拟退火等。

数值计算实践

这个问题实际上非常复杂，并且可能涉及大量的环境数据和地形数据。在现实情况下，可能需要专门的地理信息系统（GIS）软件和复杂的数据分析方法来解决。然而，为了说明问题，我们可以使用一个简化的模型和方法。

假设我们有一份数据，记录了每个高度的风力和降雪量。我们的目标是找到一个最佳的营地，即在该营地，风力和降雪量都在可接受的范围内，并且尽可能高。

扫描下方二维码可查看 Python 示例代码。使用 NumPy 和 Matplotlib 来处理这个问题，这个脚本首先生成了一些假设的风力和降雪量数据，然后找出了所有可接受的营地，并从中选择了最高的一个。最后，它绘制了风力和降雪量随高度变化的图像以及最佳营地的位置，如图 1.3 所示。

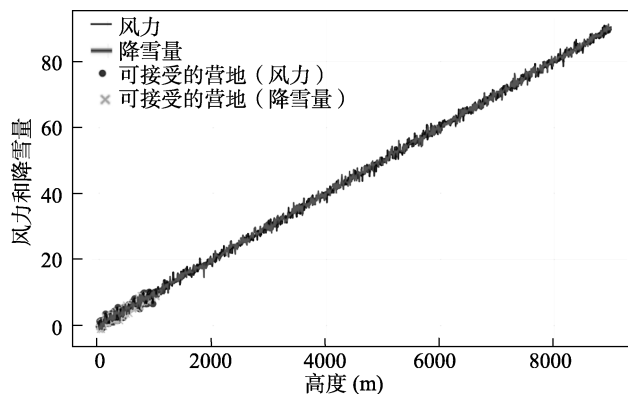


图 1.3 风力和降雪量随高度的变化及最佳营地位置



示例代码

1.9 极限在经济学中的应用简化模型和数值计算实践

经济学中有一个非常重要的概念，叫作边际效应。边际效应描述的是增加一单位的投入能够产生的效果或收益。边际效应往往随着投入的增加而递减，这也就是所谓的“边际递减效应”。

让我们以一个具体的例子来看看极限在经济学中的应用。假设一家公司生产和销售一种产品，

公司需要确定生产和销售的产品数量。公司的目标是利润最大化，而利润等于收入减去成本。

公司的收入随着销售量的增加而增加，但是每增加一单位的销售量，增加的收入（边际收入）会逐渐减少，原因是当市场上的产品过多时，公司需要降低价格来吸引消费者。

同时，公司的成本也随着生产量的增加而增加。但是每增加一单位的生产量，增加的成本（边际成本）可能会增加，也可能会减少，这取决于生产规模的经济（生产规模越大，单位成本越低）和不经济（生产规模过大，单位成本反而会增加）。

至此，公司的利润最大化问题就变成了找到使边际收入和边际成本相等的生产和销售量的问题。这个问题可以使用极限的概念来解决。具体来说，我们可以定义边际收入和边际成本为函数，然后找到使这两个函数相等的点，这个点就是利润最大化的生产和销售量。

这个案例说明了极限概念在经济学中的重要性，也展示了高等数学在解决实际问题中的重要作用。

上述的数学表达

这个问题的数学表达可以描述为找到函数的极大值或极小值的问题。假设我们有收入函数 $R(q)$ 和成本函数 $C(q)$ ，其中 q 是生产和销售数量。我们的目标是找到使最大化利润 $P(q) = R(q) - C(q)$ 实现的 q 。

首先，我们需要找到收入函数 $R(q)$ 和成本函数 $C(q)$ 的导数，也就是边际收入 $MR(q)$ 和边际成本 $MC(q)$ 。边际收入和边际成本分别表示每增加一单位生产和销售的数量，收入和成本增加的数量。在微积分中，这被称为函数的导数。

$$MR(q) = \frac{dR(q)}{dq}, \quad MC(q) = \frac{dC(q)}{dq}$$

为了利润最大化，我们需要找到使得边际收入等于边际成本的生产和销售的数量。这意味着我们需要找到满足以下条件的 q ：

$$MR(q) = MC(q)$$

在实际中，我们还需要考虑收入和成本函数的形状，可能要求解这个等式，找到满足条件的 q 。如果这个等式不能直接求解，我们可能需要用数值方法，如牛顿法来求解。

这个过程就是极限和导数在经济学中的应用。通过找到使得边际收入等于边际成本的生产和销售的数量，从而使公司实现利润最大化。

数值计算实践

我们可以使用 Python 来模拟这个问题。首先，我们定义边际成本和边际收入的函数。然后，我们计算总收入和总成本，并找到最大利润点。

扫描右侧二维码可查看 Python 示例代码，这个代码首先定义了边际成本和边际收入的函数，其次计算了总成本和总收入，再次计算了利润，并找到了最大利润点，最后它绘制了总成本、总收入和利润随生产量变化的曲线，以及最大利润点的位置，如图 1.4 所示。



示例代码

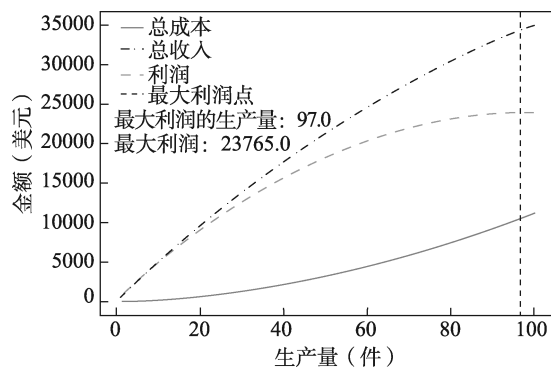


图 1.4 总成本、总收入与利润的变化曲线及最大利润点

这个例子展示了如何使用极限的概念来解决实际的经济问题。然而，这个例子非常简单，实际的经济问题可能会涉及更多的变量和更复杂的函数。

1.10 光纤传输数据的极限速度应用和简化模型及数值计算实践

光纤是现代通信网络一种重要的传输介质，它具有传输距离远、抗干扰能力强、传输速度快等优点。然而，光纤传输数据的速度并非是无无限大的，其受到一些因素的限制，并有一定的极限值。接下来，我们将探讨这个极限是如何形成的以及影响它的主要因素。

在光纤中，光信号以光脉冲的形式传输数据。每一个光脉冲代表一个比特 (bit)，即 0 或 1。理论上，光脉冲的间隔越短，传输的数据量就越大。然而，由于光脉冲间过短的间隔会导致相互干扰，也就是所谓的串扰，因此光脉冲的间隔不能无限缩小。这就形成了光纤传输数据的一个极限。

设信号的频率为 f ，光脉冲的频率为 $2f$ 。由于每个光脉冲代表 1 bit，因此数据传输速率 R (单位: bit/s) 可以表示为

$$R=2f$$

然而，在实际的光通信中，由于存在各种噪声和干扰，信号的质量并不能完全由光脉冲的频率来决定。这就需要考虑信噪比 (SNR)。在光通信中，最常用的 SNR 衡量标准是比特错误率 (bit error rate, BER)。

假设信道的信噪比为 SNR，光脉冲的能量为 E ，噪声的单边功率谱密度为 N_0 ，那么数据传输速率的理论极限可以用香农公式表示为

$$R_{\max}=f \log_2(1+\text{SNR})$$

其中， $\text{SNR} = E / N_0$ 。香农公式给出了在给定信噪比的情况下信道的最大数据传输速率，而不是实际的数据传输速率。实际的数据传输速率可能会受到各种因素的影响，如光源的稳定性、光纤的损耗、接收器的灵敏度等。

另一个影响光纤传输数据极限的因素是光纤中的衰减和色散。光在光纤中传播时会逐渐减弱，这就是光纤的衰减。而色散是指不同颜色 (即不同波长) 的光在光纤中的传播速度不同，导致光脉冲的形状和位置发生变化，从而影响数据的接收。为了避免衰减和色散的影响，必须定期设置光放大器 and 色散补偿器，但这会增加系统的复杂性和成本。

光纤传输数据的极限速度是一个由多种因素决定的极限问题。这个问题体现了极限在通信工程中的重要应用，也是一个需要运用高等数学工具解决的实际问题。深入研究这个问题可以提高光纤通信系统的性能，推动通信技术的进步。

数值计算实践

在此问题中，香农定理是关键，它告诉我们在给定信噪比的情况下信道的最大数据传输速率是多少。然而，由于这是一个理论上的极限，因此无法通过编程模拟来达到。也就是说，这个极限并不是通过调整某些参数就可以达到的，而是由物理法则决定的。实际上，当前光纤通信系统的数据传输速率还远远低于香农极限。

尽管如此，我们仍然可以使用编程和图像来直观地理解香农定理。扫描下方二维码可查看 Python 示例代码，用于绘制数据传输速率与信噪比的关系图像，在图 1.5 中，横轴表示信噪比 (线性尺度)，纵轴表示最大数据传输速率 (单位: bit/s)。可以看到，随着信噪比的增加，最大数据传输速率也在增加，但是增加的速度逐渐变慢，最终趋向于一个极限值。这个极限值就是香农极限，它由信道的带宽和信噪比决定。

这个例子只是一个理论模型，实际的光纤通信系统可能会受到很多其他因素的影响，如光纤的损耗、设备的性能、环境的干扰等，因此实际的数据传输速率低于香农极限。

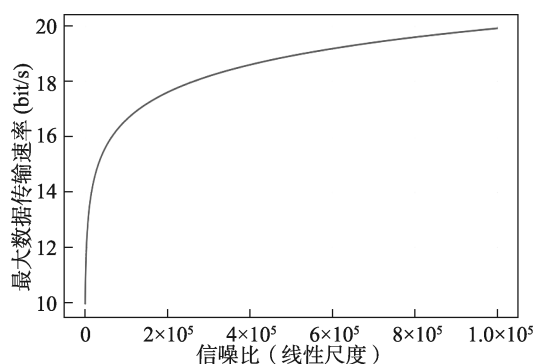


图 1.5 数据传输速率与信噪比的关系



示例代码

1.11 AI 深度学习中的极限问题应用和简化模型及数值计算实践

深度学习是人工智能 (AI) 领域的一项重要技术, 它基于神经网络进行建模, 通过对大量数据的学习和训练, 实现了语音识别、图像识别、自然语言处理等多种任务。然而, 深度学习模型的训练过程存在“梯度消失”的问题, 这是一个典型的极限问题。

在深度学习模型的训练过程中, 我们需要优化一个损失函数, 以达到模型的最佳性能。优化的方法通常是通过计算损失函数关于模型参数的梯度 (即偏导数), 然后更新模型参数以减小损失函数的值。然而, 在深度神经网络中, 由于每一层的输出都是下一层输入的函数, 因此梯度是所有层的偏导数的乘积。如果这些偏导数的绝对值都小于 1, 那么随着层数的增加, 梯度的值将趋于零, 这就是梯度消失问题。

梯度消失问题导致深度神经网络的训练变得困难, 因为当梯度接近 0 时, 模型参数的更新将非常缓慢, 甚至停止更新。为了解决这个问题, 研究者提出了多种方法, 如 ReLU 函数、初始化权重策略、正则化技术、优化器选择等。

我们以简单的全连接神经网络为例来说明梯度消失的问题。考虑一个全连接神经网络, 它由 L 层组成, 每一层都使用 sigmoid 函数作为激活函数, 代价函数是均方误差。

设 L 为神经网络的某一层, a^L 代表该层的激活值, z^L 代表该层的输入值, w^L 和 b^L 分别代表该层的权重和偏置, a^{L+1} 和 z^{L+1} 则分别代表下一层的激活值和输入值。神经网络的前向传播可以表示为

$$z^{L+1} = w^L a^L + b^L a^{L+1} = \text{sigmoid}(z^{L+1})$$

其中, sigmoid 函数的表达式为

$$\text{sigmoid}(z) = \frac{1}{1 + e^{-z}}$$

sigmoid 函数的导数可以表示为

$$\text{sigmoid}'(z) = \text{sigmoid}(z) \times [1 - \text{sigmoid}(z)]$$

假设神经网络的代价函数为均方误差:

$$C = \frac{1}{2}(y - a^L)^2$$

其中, y 是真实值; a^L 是神经网络的输出值。

代价函数关于 z^L 的梯度为

$$\left(\frac{\partial C}{\partial z}\right)^L = a^L - y$$

代价函数关于 w^L 的梯度为

$$\left(\frac{\partial C}{\partial w}\right)^L = a^{L-1} \left(\frac{\partial C}{\partial z}\right)^L$$

代价函数关于 z^{L-1} 的梯度为

$$\left(\frac{\partial C}{\partial z}\right)^{L-1} = w^L \left(\frac{\partial C}{\partial z}\right)^L \text{sigmoid}(z^{L-1})$$

可以看出,当我们计算更早层的梯度时,需要将下一层的梯度乘以权重和 sigmoid 函数的导数。如果权重和 sigmoid 函数的导数的绝对值都小于 1,那么梯度将会在反向传播的过程中迅速减小,造成梯度消失。

AI 深度学习中的梯度消失问题是一个极限问题,也是一个需要运用高等数学工具解决的实际问题。这个问题体现了极限概念在人工智能领域的重要应用,也是推动 AI 技术进步的重要问题。

数值计算实践

我们可以通过编程模拟一个多层神经网络的梯度传播过程来观察梯度消失现象。这里我们将创建一个深度全连接神经网络,并使用随机数据进行一次反向传播,然后观察每一层的梯度。

扫描下方二维码可查看代码。这个 Python 示例代码展示了在神经网络中出现的梯度消失问题。通过使用 NumPy 和 Matplotlib 库,我们定义了 sigmoid 函数及其导数,搭建了一个包含 10 层、每层 50 个节点的深度神经网络。利用随机初始化的权重和激活值进行前向传播和反向传播。反向传播过程存储了每一层的梯度。通过绘制图表,以对数缩放的方式展示不同层中的平均梯度值变化情况,从而呈现神经网络中梯度逐渐消失的现象。

在图 1.6 中,横轴表示网络的层数,纵轴表示梯度绝对值的均值。可以看到,从输出层到输入层,梯度的值迅速减小,表现出了梯度消失的现象。这是因为在每一层,梯度都与权重和 sigmoid 函数的导数相乘,而这两个因子的值都小于 1,因此梯度在反向传播过程中迅速减小。

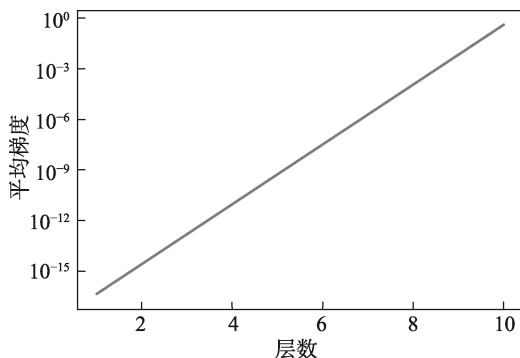


图 1.6 神经网络中出现的梯度消失问题



示例代码

1.12 移动通信中的信号衰减应用和简化模型及数值计算实践

移动通信技术在现代生活中扮演着重要的角色,它依赖无线电波来传输语音、文本、图片和视频等数据。然而,当无线电波在空气中传播时,其能量会随着距离的增加而逐渐减少,这种现象被称为信号衰减。

信号衰减的速率用数学模型来描述,这个模型通常包括一个或多个参数,如距离、频率和环境

条件等。一般情况下，信号的功率密度会按照距离的平方的倒数进行衰减，这被称为自由空间路径损耗模型。然而，在实际的无线通信系统中，信号的衰减速度通常比这个模型预测的要快，这是因为有很多其他因素会导致额外的信号损耗，如建筑物、地形、大气和天气条件等。

信号衰减模型的一个重要应用是无线通信网络的设计和优化。例如，移动通信网络的基站布局需要考虑到信号的覆盖范围和衰减特性，以确保用户可以在网络覆盖区内获得满意的通信服务。此外，无线通信设备的功率控制也需要考虑信号的衰减特性，以达到有效的能源利用和降低干扰。

这个问题可以转化为一个优化问题。假设我们有 N 个可能的基站位置、 M 个用户，我们的目标是 minimized 总的传播损失和建设成本，同时满足每个用户的服务需求。

假设 x_i 表示是否在第 i 个位置建立基站，它是一个二进制变量，如果在该位置建立基站则 $x_i = 1$ ，否则 $x_i = 0$ 。 y_{ij} 表示第 i 个基站是否服务第 j 个用户，它也是一个二进制变量，如果第 i 个基站服务第 j 个用户则 $y_{ij} = 1$ ，否则 $y_{ij} = 0$ 。 L_{ij} 表示第 i 个基站到第 j 个用户的信号传播损失， C_i 表示在第 i 个位置建立基站的成本。

我们的目标函数可以表示为

$$\min \sum_{i=1}^N C_i x_i + \sum_{i=1}^N \sum_{j=1}^M L_{ij} y_{ij}$$

其中

$$\sum_{i=1}^N y_{ij} \geq 1 (j=1, \dots, M)$$

这个问题是一个混合整数优化问题，可以通过现有的优化工具进行求解。

至于无线通信设备的功率控制问题，设 p_i 表示第 i 个基站的发送功率， g_{ij} 表示第 i 个基站到第 j 个用户的信号增益， n_j 表示第 j 个用户的噪声功率，我们的目标是使得所有用户的信噪比满足一定的阈值 T ：

$$p_i \frac{g_{ij}}{n_j} \geq T (i=1, \dots, N, j=1, \dots, M)$$

这也是一个优化问题，可以通过调整每个基站的发送功率 p_i 来求解。

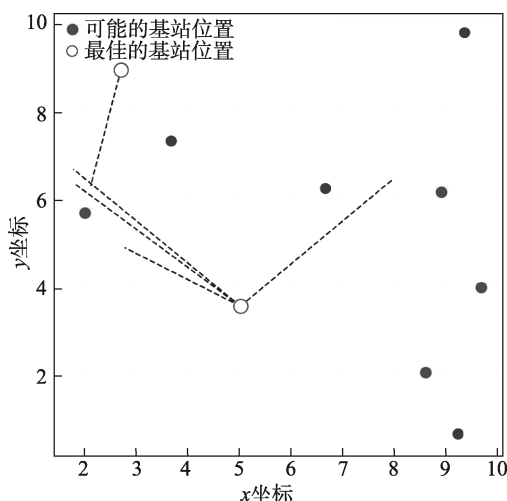
极限在信号衰减的描述和分析中起到了关键作用。当距离趋于无穷大时，信号的功率密度将趋于 0，这就是一个典型的极限问题。通过理解和应用极限的概念，我们可以更准确地描述和预测信号的衰减特性，从而优化无线通信网络的性能。

数值计算实践

以上所描述的问题都是优化问题。对于第一个问题，我们可以使用整数线性规划 (Integer Linear Programming, ILP) 进行求解。对于第二个问题，我们可以通过求解非线性规划问题来确定每个基站的功率。

由于这些问题都涉及求解优化问题，因此在实际的编程实践中，我们需要借助专门的优化库，如 Python 的 SciPy 库或 CVXPY 库。这些问题的求解都涉及复杂的数学运算，以及对优化理论的深入理解。

扫描下方二维码可查看代码。这个 Python 示例代码使用 CVXPY 库来解决基站位置优化问题。假设有一些可能的基站位置和用户，以及基站的建设成本和信号传播损失。通过二进制变量 x 表示是否在每个位置建立基站，二维二进制变量 y 表示每个基站是否服务每个用户。程序使用 CVXPY 库定义了目标函数、约束条件和求解问题。在绘图部分，实心圆圈表示可能的基站位置，而空心圆圈表示最佳的基站位置，如图 1.7 所示。程序输出了最优值，即最佳变量 x 和 y 的取值。



示例代码

图 1.7 可能的基站位置及最佳的基站位置

1.13 自动驾驶技术中传感器的感知范围应用和简化模型及数值计算实践

在自动驾驶技术中，传感器如雷达、激光雷达、摄像头等，是非常重要的组成部分。它们向车辆提供了对环境的感知能力，帮助车辆识别和跟踪周围的物体，如其他车辆、行人、路标等。传感器的感知范围，即传感器可以有效检测到物体的最大距离，是一个关键参数。在此背景下，我们可以提出一个典型的极限问题：当物体与传感器的距离越来越大时，传感器能否有效地检测到物体？

为了解决这个问题，我们需要了解传感器的工作原理。雷达和激光雷达都是通过发射电磁波或激光脉冲然后接收反射回来的信号来检测物体的。由于传播的信号会随着距离的增加而衰减，因此当物体与传感器的距离足够大时，反射回来的信号可能会低于传感器的检测阈值，导致无法有效地检测到物体。这就是一个典型的极限问题：当距离趋于传感器的最大感知范围时，反射回来的信号强度趋于传感器的检测阈值。

在实际的自动驾驶系统中，这个极限问题需要通过设计和优化传感器的参数以及处理反射信号的算法来解决。例如，提高传感器的发射功率可以增加反射回来的信号强度，从而增大感知范围；优化信号处理算法可以提高检测的灵敏度，从而在低信号强度下仍然能够有效地检测到物体。

设 d 是感知范围， P 是传感器的发射功率， A 是接收到的信号强度，我们需要找到最优的 P 使得 A 保持在一个可以接受的阈值 T 以上，从而确保感知范围 d 的最大化。

这个优化问题的数学形式为

$$\begin{aligned} & \max d \\ & \text{s.t. } \frac{P}{d^2} \geq T \end{aligned}$$

这是一个约束优化问题，它反映了信号强度 A 随着距离 d 平方的增大反向减小的物理定律（称为反距离平方定律）。

此外，为了提高在低信号强度下的检测能力，我们可以考虑使用阈值函数（threshold function）。假设 x 是接收到的信号强度，我们可以定义一个阈值函数 $f(x)$ ，当 x 大于一个给定的阈值 T 时， $f(x)$ 返回 1，表示物体被检测到；当 x 小于 T 时， $f(x)$ 返回 0，表示物体没有被检测到。

$$f(x) = \begin{cases} 1, & x \geq T \\ 0, & \text{其他} \end{cases}$$

选择合适的阈值 T 是关键，它需要权衡检测率（即正常情况下物体被正确检测出的概率）和误报率（即没有物体但是被错误检测出的概率）。这可以通过 ROC 曲线（接受者操作特征曲线）来进行选择，ROC 曲线描述了检测率和误报率之间的关系。

数值计算实践

这是一个求解非线性规划问题的例子，可以使用 Python 的 SciPy 库进行求解。扫描下方二维码可查看代码。

首先，定义了一个目标函数和一个约束条件。目标是最大化目标函数，而约束条件则涉及目标函数中的参数。接着，使用 minimize 函数求解问题，并输出最优值以及最优变量。可视化部分绘制了目标函数和约束条件随参数变化的图形。图 1.8 (a) 显示了目标函数值随参数变化的趋势，并标出了最优解。图 1.8 (b) 显示了约束函数值与参数的关系，并在约束条件界限处标出了虚线。

得出：

最优值：3.1622776382067648

最优变量 d : [3.16227764]

这里，我们首先在 0.1 到 10 之间生成了一些 d 的值，然后计算了在这些 d 下目标函数和约束函数的值。最后，我们分别画出了目标函数和约束函数的变化趋势，并在图中标出了最优解，如图 1.8 所示。



示例代码

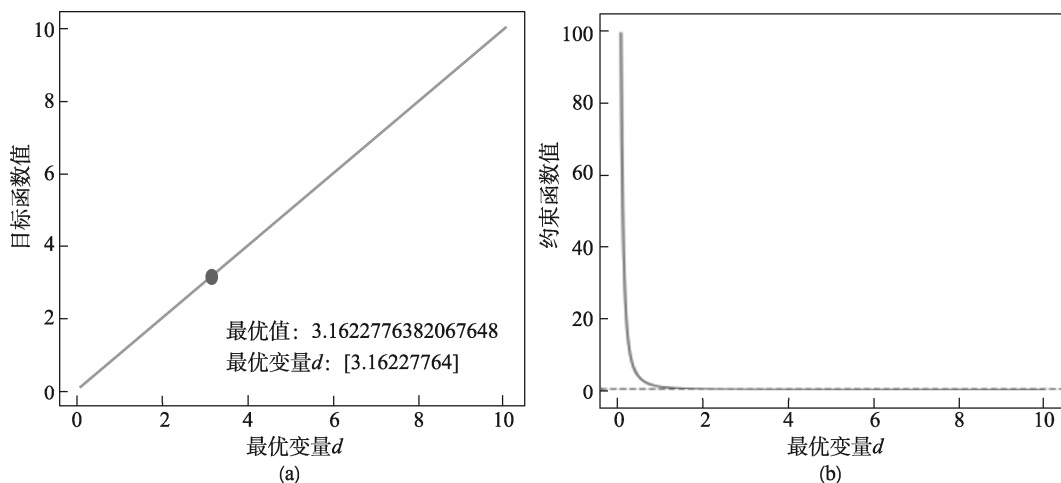


图 1.8 目标函数和约束条件的变化趋势与最优解

1.14 极限在化学反应速率中的应用和简化模型及数值计算实践

化学反应速率是反应过程中反应物质的消耗速度或产物的生成速度。在许多实际场景中，化学反应的速率可能会随着反应的进行而改变，这就涉及极限的概念。

一个常见的例子是一阶反应。一阶反应的反应速率与反应物的浓度成正比，随着反应的进行，反应物的浓度逐渐减小，反应速率也会逐渐降低。在这个过程中，我们可以定义一个极限情况，即当反应物的浓度趋近于 0 时，反应速率也会趋近于 0。

一阶反应的反应速率 v 被定义为

$$v = k[A]$$

其中, v 是反应速率; k 是反应速率常数; $[A]$ 是反应物 A 的浓度。

因此, 反应速率与反应物 A 的浓度直接相关。

随着反应的进行, 反应物 A 的浓度逐渐减小, 所以反应速率也会逐渐降低。根据一阶反应的规律, 反应物 A 的浓度以指数方式减小, 可以用下面的公式表示:

$$[A] = [A_0]e^{-kt}$$

其中, $[A_0]$ 是反应开始时刻反应物 A 的浓度; t 是反应时间。

当反应进行到一定程度时, $[A]$ 会趋近于 0, 根据反应速率公式, 此时的反应速率 v 也会趋近于 0。这就是一阶反应的一个极限情况。

极限概念在这里的应用有两个方面:

(1) 在实际应用中, 我们可以通过这种方式预测反应的结束时间。当反应速率趋近于 0 时, 我们可以认为反应已经基本完成。这在化学制药、材料合成等领域有重要应用。

(2) 在理论研究中, 这种极限分析的方法可以帮助我们理解反应机理, 预测反应过程, 并为优化反应条件、提高产率等提供理论依据。

这个案例展示了极限在化学反应研究和工程实践中的应用。

数值计算实践

这个问题主要涉及反应动力学, 我们可以使用 Python 来模拟一阶反应过程并进行可视化展示。扫描下方二维码可查看代码。

在这个 Python 示例代码中, 首先定义了一阶反应的参数, 包括反应速率常数 k 和初始反应物浓度 A_0 。其次, 定义了反应进行的时间, 从 0 到 20 单位时间。再次, 根据一阶反应的公式计算了反应物的浓度 $[A]$ 和反应速率 v 。最后, 绘制了反应物浓度和反应速率随时间变化的图像, 如图 1.9 所示。

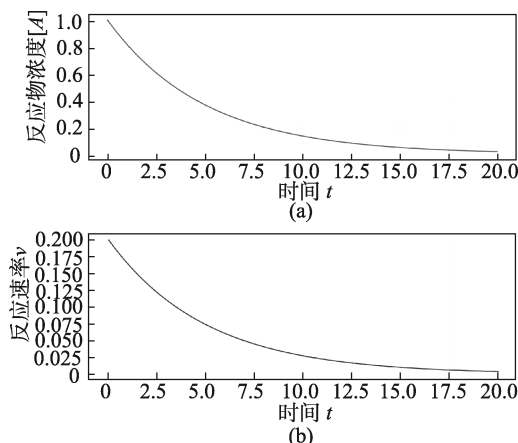


图 1.9 反应物浓度与反应速率随着时间变化的曲线

可以看到, 随着时间的推移, 反应物的浓度逐渐降低, 反应速率也逐渐降低, 最终趋向于 0。这就是一阶反应中的极限情况。

1.15 极限在物理中的应用和简化模型及数值计算实践

在物理学中, 极限概念的应用十分广泛, 其中一个显著的例子就是在热力学中寻找绝对零度。绝对零度, 定义为 0 K, 是热力学温度的最低可能值, 它代表着一个理论极限——在这个温度下, 粒子的热运动几乎停止。

然而, 实际上, 绝对零度是一个不可能实现的极限。这涉及第三定律, 也就是无法在有限的步

骤内达到绝对零度的热力学定律。任何冷却过程都会使系统趋近于这个极限，但无论如何都无法完全达到。

物理学家使用各种冷却技术，例如激光冷却和稀有同位素稀释制冷等，可以将物质冷却到仅比绝对零度稍高的温度。这些冷却技术的目标就是尽可能接近这个极限。极限的概念在这里提供了一个理论框架，帮助科学家理解他们的实验结果和设计更有效的冷却策略。

激光冷却是一种使用激光的辐射压力来冷却原子或分子的技术，已经能够将气体冷却到接近绝对零度，其基于多普勒效应和辐射力的基本原理。

假设一个原子处于速度为 v 的运动状态，当它与一个频率为 ν 的、与其相向而行的光子相互作用时，由于多普勒效应，原子感受到的光子频率被“红移”到 $\nu' = \nu(1 + v/c)$ ，其中 c 是光速。如果 ν' 接近原子的共振频率 θ_0 （即原子能量级间的跃迁频率），那么原子有可能吸收这个光子，并在吸收光子后获得一个反向的冲量。之后，原子以一定的概率通过自发辐射将能量释放，自发辐射的方向是随机的，因此在多次吸收和释放光子后，原子的总体动能减小，也就是说原子被冷却了下来。

具体的数学表达可以使用一维情形简化，假设原子在 x 轴上运动，辐射力 F 可以表示为

$$F = hk(\Gamma/2)\Delta / (\Delta^2 + \Gamma^2/4)$$

其中， h 是约化普朗克常数； k 是激光波数，与激光频率有关； Γ 是原子的自然线宽； $\Delta = k\nu - \nu_0 + v$ 是原子和激光之间的失调量。

当原子的速度增大时，失调量 Δ 增大，辐射力 F 减小，这使得原子速度减小，温度降低，这就是激光冷却的基本原理。

这个案例表明，极限不仅是数学中的基本概念，其在助力理解自然界的基本法则和探索科技前沿方面也起着关键的作用。

数值计算实践

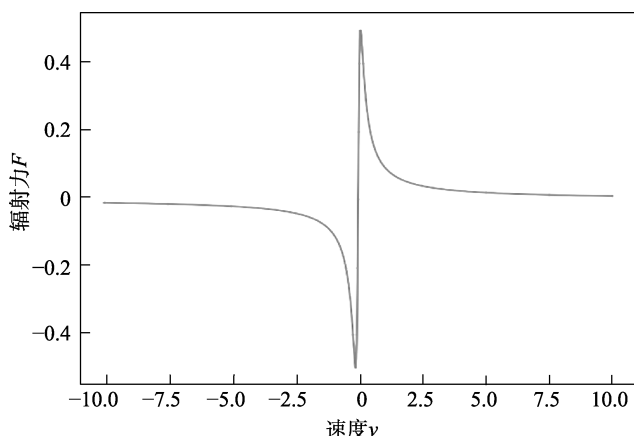
在这个问题中，我们需要计算辐射力 F 对原子速度 v 的影响。在这里，我们假设激光的频率 ν 是固定的，而原子的速度 v 是变化的。以下是一个 Python 示例代码来绘制辐射力 F 随原子速度 v 变化的图像。扫描下方二维码可查看代码。

在这段 Python 示例代码中，我们首先定义了相关的物理常数，包括约化普朗克常数 h （或 h_{bar} ）、激光波数 k 、原子的自然线宽 γ 、激光频率 c 、原子的共振频率 θ_0 。

接下来，我们定义了一个函数 `radiation_force` 来计算辐射力 F 。这个函数接收一个参数，即原子的速度 v ，然后计算辐射力 F 。

然后，我们定义了一个原子速度的范围， $-10 \sim 10$ ，总共 1000 个点。

最后，我们计算了辐射力 F ，并绘制了辐射力 F 随原子速度 v 变化的图像，如图 1.10 所示。



示例代码

图 1.10 辐射力随原子速度的变化曲线

这个图形可以帮助我们理解激光冷却的基本原理：当原子的速度增大时，辐射力 F 减小，从而使得原子速度 v 减小，实现冷却效果。

1.16 极限在生物学中的应用和简化模型及数值计算实践

在生物学中，极限的概念常被应用于种群动态的研究，尤其是种群增长模型。在理想的环境下，一个物种的数量可能会按照指数函数无限地增长。但实际上，环境资源和空间总是有限的，因此，真实的种群增长过程通常会受到一种或多种因素的限制，最终趋向于一个稳定的种群大小，这就是所谓的环境承载力。

一个常用的模型是逻辑增长模型 (Logistic Growth Model)，它考虑了环境承载力对种群增长的影响。在这个模型中，种群的增长速度随着种群大小的增加而减小，直到达到一个极限值，这个极限值就是环境承载力。在这个点上，种群的出生率和死亡率达到平衡，种群的规模不再增长。

该模型可以用微分方程来表示：

$$\frac{dN}{dt} = rN \left(1 - \frac{N}{K} \right)$$

其中， N 是种群规模， r 是种群的固有增长率， K 是环境承载力。可以看到，当 N 接近 K 时，种群的增长率 $\frac{dN}{dt}$ 接近 0，这就是一个典型的极限的例子。

这个案例显示了极限在生物学中的应用，帮助我们理解种群的动态和复杂性，并能为保护生物多样性和可持续发展提供理论支持。

数值计算实践

对于逻辑增长模型，我们可以使用 Python 的 SciPy 库中的 `odeint` 函数来解这个微分方程。扫描下方二维码可查看代码。下面的 Python 示例代码首先定义了微分方程 `logistic_growth`，这个函数接收当前的种群规模 N 、当前时间 t 、固有增长率 r 和环境承载力 K ，返回种群的增长率。

接下来，使用 SciPy 库中的 `odeint` 函数求解微分方程。`odeint` 函数接收微分方程、初始条件和时间范围，返回微分方程的解。

最后，绘制了种群规模 N 随时间 t 变化的图像，如图 1.11 所示。从图中可以看到，随着时间的推移，种群规模逐渐增大，最终趋于一个稳定值，这个值就是环境承载力 K 。

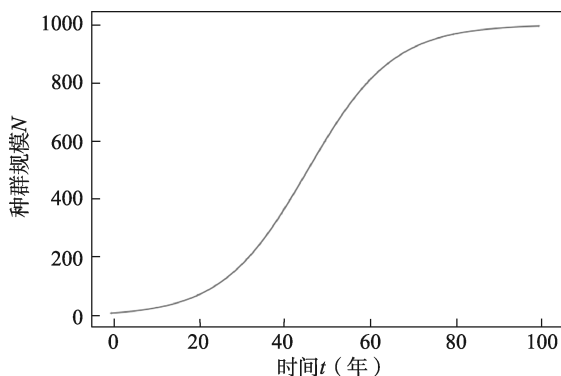


图 1.11 种群规模 N 随时间 t 变化的曲线



示例代码

这个图形可以帮助我们理解逻辑增长模型和极限在生物学中的应用。

1.17 极限在工程设计中的应用和简化模型及数值计算实践

在工程设计中，构造物的承重能力是极其重要的考量因素，大到桥梁、建筑、航空器，小到桌椅，这些物体的设计和建造都需要充分计算其极限承重。极限承重的概念可以被看作物体所能承受而不导致其破裂或失去功能的最大负荷。

考虑一座桥梁，桥梁的设计和建造都必须考虑其能承受的最大重量，这就涉及极限的概念。如果桥梁的承重超过了这个极限，桥梁就会损坏甚至崩塌，对人们的生命和财产安全构成威胁。

极限状态包括两种：

承载力极限状态：这是指结构不能承受超过设计荷载，否则会导致坍塌或者结构失效。例如，一座桥梁在设计时就会考虑最大的荷载（如车辆荷载、风荷载、地震荷载等），并确保桥梁在这些最大荷载下仍能保持稳定。

服役极限状态：这是指在正常使用荷载下，结构不能满足使用功能，例如过大的变形或振动等。

在数学上，我们可以根据荷载和材料强度等因素，使用静力学和材料力学的方程来计算和预测结构的极限状态。具体到桥梁设计上，一般会使用有限元分析或其他结构分析方法，结合材料的应力-应变关系，以及荷载和约束条件，建立相应的数学模型和方程组，来预测和设计桥梁的极限状态。

这个案例向我们展示了极限在工程设计中的应用，并强调理解和使用极限概念在现代社会中的重要性。

数值计算实践

我们可以使用一个简化的例子来理解极限在结构工程中的应用。考虑一个简单的悬臂梁，其一端固定，另一端受到垂直荷载 P 的作用。梁的挠度（即梁的最大位移或弯曲）可以用以下公式来计算：

$$D = \frac{PL^3}{3EI}$$

其中， D 是梁的最大挠度； P 是作用在梁上的荷载； L 是梁的长度； E 是梁材料的弹性模量； I 是梁的面积惯性矩。

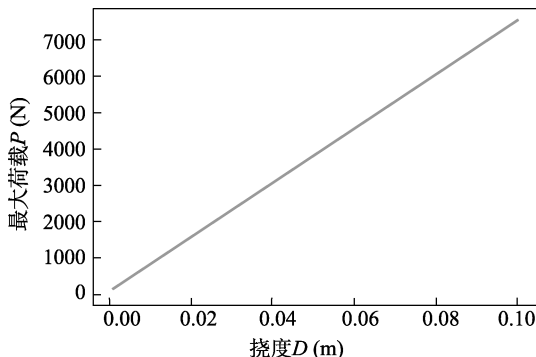
现在假设我们已经知道梁的长度、材料弹性模量、面积惯性矩以及梁的最大允许挠度。我们想知道梁可以承受的最大荷载是多少。

这个问题可以通过解上述方程求得：

$$P = \frac{3DEI}{L^3}$$

首先我们可以创建一个函数来计算最大荷载 P ，然后我们可以改变挠度 D 的值并且绘制出 P 随 D 变化的曲线。这里我们假设梁的长度 L 、材料弹性模量 E 和面积惯性矩 I 是已知的。

在图 1.12 中，你可以看到挠度 D （即梁的最大位移或弯曲）与梁可以承受的最大荷载 P 之间的关系。当挠度 D 增大时，梁可以承受的最大荷载 P 也随之增大。扫描下方二维码可查看代码。



示例代码

图 1.12 荷载随挠度变化的曲线

这个例子虽然很简单，但它展示了如何使用极限的概念来进行结构设计和分析。在实际应用中，工程师会考虑更多的因素，比如梁的形状、材料的非线性行为、动态荷载等，这会使问题变得更复杂，但基本的思路是相同的。

1.18 大气压强与海拔高度关系的应用和简化模型及数值计算实践

大气压强与海拔高度之间的关系体现了极限概念的实际应用。海拔越高，大气压强越小，这是因为海拔高度增加，大气层的厚度减小，重叠的气体分子减少，因此大气压强降低。实际上，当海拔高度趋于无穷大时，大气压强将趋于 0，这就是极限的概念。

我们可以通过数学公式来描述这种关系。大气压强 P 与海拔高度 h 的关系可以通过下式表示：

$$P = P_0 \exp(-Mgh / RT)$$

其中， P_0 是海平面的大气压强； M 是空气的平均摩尔质量； g 是重力加速度； R 是理想气体常数； T 是温度。这个公式就描述了大气压强随着海拔高度增加而呈指数衰减的现象。

这种关系在许多领域都有重要应用，如气象学、航空、登山等。例如，在航空领域，飞机在飞行过程中需要考虑大气压强的变化，以确定最佳的飞行高度。在登山领域，登山者需要了解大气压强随着海拔高度的变化，以便预防高原反应。

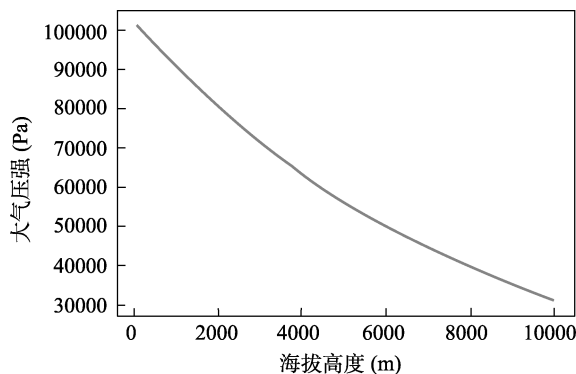
通过这个案例，我们可以看到极限在自然科学中的实际应用，同时也可以了解到掌握极限的概念，帮助我们更好地理解 and 解决实际问题。

数值计算实践

这个模型假设温度 T 是常数，即不随海拔高度改变。在实际情况下，温度确实会随着海拔的升高而减小，但为了简化问题，我们这里假设它是恒定的。在进行更复杂的模型计算时，我们需要考虑这个因素。

现在，我们可以使用 Python 的 Matplotlib 库来创建一个简单的模型，来展示大气压强如何随海拔的变化而变化。

扫描下方二维码可查看代码。在这段 Python 示例代码中，首先定义了所需的各种常数，然后计算了海拔高度从 0~10000 m 时的大气压强，最后用图形展示了大气压强如何随着海拔高度的增加而减小，如图 1.13 所示。随着海拔高度的升高，大气压强确实在减小，并且在海拔高度非常高时，大气压强接近于 0，这就体现了极限的概念。



示例代码

图 1.13 大气压强随海拔高度变化的曲线

1.19 噪声信号过滤器的设计应用和 简化模型及数值计算实践

在信号处理中，尤其是在通信系统、音频处理和图像处理等领域，噪声信号过滤是非常重要的。它的主要目标是从噪声中提取出我们想要的信号，这就需要有一个噪声过滤器。在设计噪声过滤器时，我们需要考虑极限的概念。

过滤器的主要工作原理是通过调整频率响应来削弱或强化某些频率的信号。理想的过滤器会完全阻止不需要的频率，并且不会影响需要的频率。然而，现实中的过滤器无法做到这一点。它们有一定的转换带宽，在这个带宽内，信号的频率响应会逐渐从通频转变为阻频。

在设计过滤器时，我们通常会设置一个截止频率，这是过滤器开始显著衰减信号的频率。在这个频率之前，过滤器对信号的影响很小；在这个频率之后，过滤器对信号的衰减就会显著增加。实际上，当我们增加截止频率，信号的衰减程度将趋向于无穷大，这就体现了极限的概念。

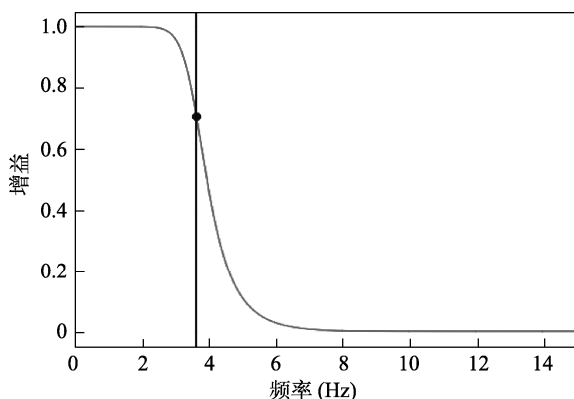
通过这个案例，我们可以了解到极限的概念在信号处理中的实际应用，掌握极限的概念，可以帮助我们更好地设计和优化噪声过滤器。

数值计算实践

在信号处理中，最常用的过滤器是低通滤波器（LPF），它可以通过去除高频信号来削弱噪声。我们可以使用 Python 中的 SciPy 库来实现一个简单的数字低通滤波器。

在这个 Python 示例代码中，首先必要的 NumPy、SciPy 信号处理模块和 Matplotlib 库被导入，用于数值计算、滤波器设计、频率响应计算以及绘图。接着，定义 `butter_lowpass` 函数，根据给定截止频率、采样率和阶数来计算 Butterworth 低通滤波器的系数。然后，`butter_lowpass_filter` 函数用设计好的滤波器对输入信号进行滤波操作，通过 `filter` 函数进行实际的滤波。在设定参数阶段，设置滤波器的阶数、采样率和截止频率。接下来，使用 `butter_lowpass` 函数获取滤波器系数，再利用 `freqz` 函数计算频率响应的幅度和相位信息。频率响应图生成步骤中，Matplotlib 被用来绘制滤波器的增益幅度响应曲线，标记截止频率、增益的特定点，垂直线用于表示截止频率。最后，设定图表界限、标题和坐标轴标签，并添加网格等装饰，绘制得到的频率响应图被显示出来。扫描下方二维码可查看代码。

我们生成的低通滤波器频率响应如图 1.14 所示。可以看到，在截止频率之前，滤波器对信号的影响很小（即增益接近 1）；在截止频率之后，滤波器对信号的衰减就会显著增加（即增益迅速下降）。这就是过滤器极限性质的可视化表示。



示例代码

图 1.14 低通滤波器的频率响应

1.20 极限在市场经济中的应用和简化模型及数值计算实践

在经济学中，“市场饱和”是一个经常被讨论的概念。当一个市场的所有潜在消费者都已经获得了某个产品或服务，或者他们对该产品或服务的需求已经被满足，我们就说这个市场已经“饱和”。在这种情况下，销售增长将会达到一个上限。

考虑一个新产品在市场上的推广。一开始可能只有少数消费者知道并购买了该产品。随着产品的推广和消费者对产品认知度的提高，产品的销售量可能会急剧增加。然而，当所有可能的消费者都已经知道并尝试了这个产品后，产品的销售增长就可能放缓，最终达到一个极限，这就是市场饱和。

这个过程可以通过数学模型来描述，其中最常用的是 S 曲线或逻辑函数。这个模型的数学形式为

$$S(t) = \frac{K}{1 + e^{-r(t-t_0)}}$$

其中， $S(t)$ 是销售量， t 是时间， K 是市场饱和度（即最大销售量）， r 是增长速率， t_0 是增长的起始时间。在这个模型中，随着时间的推移，销售量 $S(t)$ 将趋向于极限 K 。

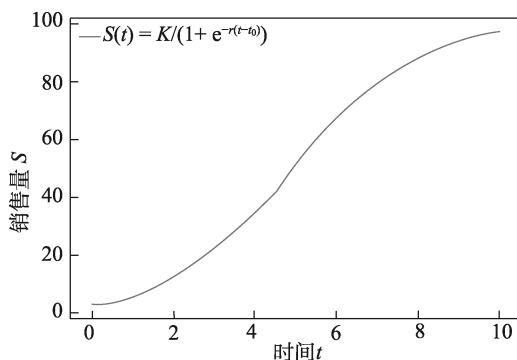
通过这个案例，我们可以看到极限概念在市场经济中的实际应用，它不仅可以帮助我们理解市场饱和的过程，而且可以帮助我们制定未来的销售策略。

数值计算实践

这是一个实现 S 曲线或逻辑函数并进行可视化展示的 Python 示例代码，使用 NumPy 和 Matplotlib 库生成并绘制了一个 S 曲线函数。首先，定义 S 曲线函数，其中包含了参数 K （最大销售量）、 r （增长速率）和 t_0 （增长的起始时间）。接着，生成一个时间序列 t ，在每个时间点计算对应的销售量，并将结果绘制为图形。扫描下方二维码可查看代码。

在这个 Python 示例代码中，我们定义了一个 S 曲线函数 `s_curve`，它表示销售量随时间的变化。我们设定了市场饱和度 $K = 100$ 、增长率 $r = 0.7$ 和增长的起始时间 $t_0 = 5$ 。然后我们生成了一个时间序列 t ，并计算了每个时间点的销售量 `sales`。最后，我们绘制了销售量随时间变化的变化曲线，如图 1.15 所示。

从这个图中可以看到，随着时间的推移，销售量逐渐增加，最终趋于一个极限值 K ，这就是市场饱和的过程。



示例代码

图 1.15 销售量的变化曲线

1.21 极限在制药中的应用和简化模型及数值计算实践

在药物开发和生产中，药物的溶解速率是一个非常重要的参数，它决定了药物在人体内的吸收速率和效率。药物溶解速率的测量是药物生产中一个关键的步骤，它的极限状态，即药物溶解的

极限速率，是药品研发和质量控制的重要数据。

考虑一个典型的药物溶解实验。药物被放入一个含有溶解介质（例如水或某种溶剂）的容器中。随着时间的推移，药物会逐渐溶解，药物的浓度会逐渐增加，直到达到一个稳定值，这个稳定值就是药物溶解的极限速率。

这个过程可以用数学模型来描述，最常用的是 Noyes-Whitney 方程，它描述了药物溶解速率 $\frac{dM}{dt}$ （ M 为溶解药物的质量）与药物表面积（ A ）、溶剂和药物之间的饱和溶液浓度差（ $C_0 - C$ ），以及溶剂对药物的扩散系数（ D ）的关系：

$$\frac{dM}{dt} = DA(C_0 - C) / h$$

其中， h 是扩散层厚度。

通过上述公式我们可以看到，药物的溶解速率与药物的表面积、溶剂和药物之间的饱和溶液浓度差、溶剂对药物的扩散系数有关，这些都是药物溶解的重要参数。当药物全部溶解的那一刻，溶解速率达到一个极限值，这个极限值是药物溶解的最大速率。

极限在制药中的应用是非常广泛的，它可以帮助我们理解药物的溶解过程，并为药物的开发和生产提供重要的数据支持。

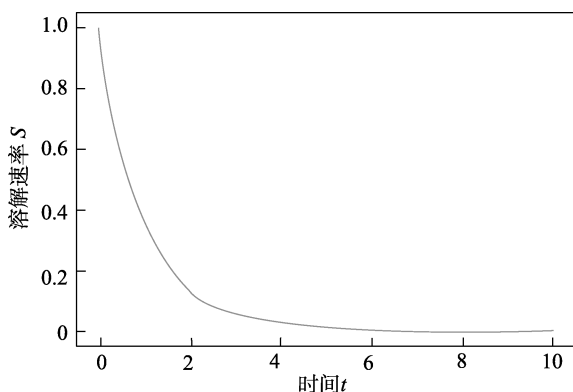
数值计算实践

在实际的制药和药物研究中，科研人员会使用专门的软件和工具，比如 MATLAB、Python 的 `scipy.integrate` 等库来求解这样的微分方程，并利用实验数据来验证和优化模型。

我们先用一个简化的模型来模拟药物溶解的过程。我们假设药物的溶解过程是一个指数递减的过程，即随着时间的增加，药物溶解的速率会逐渐减小。这可以用一个简单的指数函数来描述。扫描下方二维码可查看代码。

在这个 Python 示例代码中，使用 NumPy 和 Matplotlib 库来生成和绘制药物溶解速率随时间变化的曲线。首先，定义了药物溶解速率函数，其中包含参数 k ，表示速率的系数。然后，生成一个时间序列 t ，在每个时间点计算对应的药物溶解速率，并将结果绘制为图形，如图 1.16 所示。图中呈现了药物溶解速率随时间变化的趋势，横轴表示时间，纵轴表示药物溶解速率。

图 1.16 显示了随着时间的推移，药物溶解的速率逐渐减小，最终趋向于 0。这就是一个极限的概念。虽然这个模型很简单，但它可以带给我们一个关于药物溶解过程和极限概念的直观理解。



示例代码

图 1.16 药物溶解速率的变化曲线

1.22 极限在气候模型中的应用和简化模型及数值计算实践

气候模型是一种复杂的数学模型，用于模拟和预测地球气候系统的行为。这些模型需要考虑许

多因素，包括大气、海洋、陆地和冰川之间的相互作用，以及太阳辐射、地球的自转和倾斜等外部驱动因素。

在这些模型中，极限的概念经常出现。例如，考虑气候模型中的一个关键因素——全球温度。随着温室气体排放的增加，地球的平均温度正在上升。但是，地球温度的升高并不是无限的，它受到多种因素的制约，如地球表面反射太阳辐射的能力（被称为地球的反照率）以及大气中二氧化碳的吸收和释过程等。

在这种情况下，我们可以考虑温度升高的极限状态，这是当温室气体排放持续增加并达到一个极高水平时的状态。在这个极限状态下，地球的温度会达到一个上限，超过这个上限，地球的气候系统将发生不可逆的变化，可能导致灾难性的后果。

此外，极限的概念还可以用来考虑气候模型的不确定性。由于气候系统的复杂性，气候模型通常包含许多不确定性。在这种情况下，极限可以帮助我们定义模型预测的最大和最小范围，从而更好地理解模型预测的不确定性。

模拟地球气候系统的主要方法是使用气候模型，这是一种复杂的数学模型，包括大量的微分方程，描述了大气、海洋、冰川和陆地等各个部分的物理过程。

为了模拟温度升高的极限状态，我们可以设定一种情况，即温室气体的排放持续以一定的速率增加。这可以用数学表达式来表示，例如 $E(t) = E_0(1+r)^t$ ，其中 $E(t)$ 是 t 时刻温室气体的排放量， E_0 是初始排放量， r 是排放增长率。

然后，我们将这个排放情况作为输入条件，代入气候模型进行模拟，计算出在这种情况下地球的平均温度随时间的变化。这个过程可以用一个高阶微分方程来表示，例如：

$$\frac{dT}{dt} = f(E(t), T(t), P)$$

其中， $T(t)$ 是 t 时刻地球的平均温度， P 表示其他影响气候的参数，如太阳辐射、地球的反照率等。

模型的解（即温度随时间的变化）将会显示一个极限状态。然后我们可以通过调整排放增长率 r 来探索如何避免这个极限状态，这就是所谓的减排目标。

因此，极限在气候模型中的应用是非常重要的，它可以帮助我们理解和预测地球气候系统的行为，并为应对全球气候变化提供重要的参考依据。

数值计算实践

我们可以使用一个简化的模型来模拟温室气体排放和地球温度之间的关系。我们可以假设温度的增加是与温室气体排放量是成正比的。这可以用下面的简单线性模型来描述。扫描下方二维码可查看代码。

在这个 Python 示例代码中，使用 NumPy 和 Matplotlib 库来模拟和绘制随时间变化的温室气体排放量和地球平均温度。首先，定义了温室气体排放量随时间变化的函数和地球平均温度随时间变化的函数，其中包括初始排放量、排放增长率和温度与排放量的比例系数。然后，生成一个时间序列 t ，在每个时间点计算对应的温室气体排放量和地球平均温度，并将结果绘制为两个图形。第一个图形展示了温室气体排放量随时间的变化，横轴表示时间 (t)，纵轴表示排放量 (E)。第二个图形展示了地球平均温度随时间的变化，横轴同样表示时间，纵轴表示地球平均温度 (T)。两个图形分别展示了排放量和温度随时间变化的趋势，如图 1.17 所示。

这个模型非常简单，无法反映真实气候系统的复杂性。但是，它可以给我们一个直观的理解，即如果温室气体的排放量持续增加，地球的平均温度也会持续上升。这是一种极限状态，如果不加以控制，可能会引发严重的气候问题。



示例代码

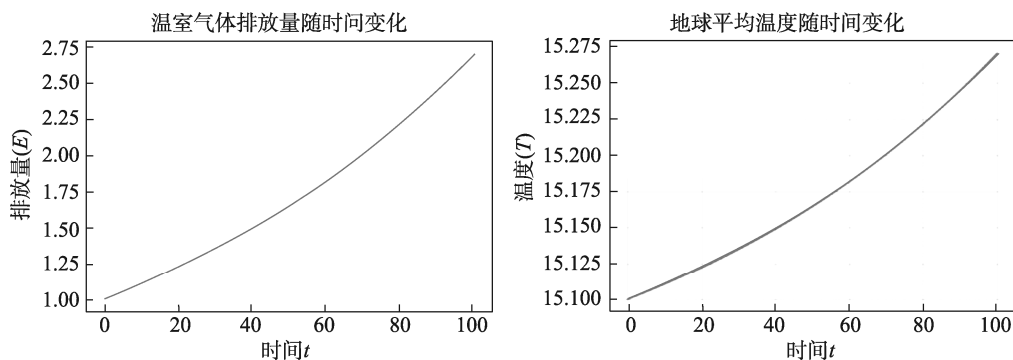


图 1.17 温室气体排放量与地球平均温度随时间的变化趋势

1.23 极限在飞机机翼设计中的应用和简化模型及数值计算实践

在飞机设计中，机翼是至关重要的部分。机翼的设计和制造必须满足多个条件和标准，其中之一就是承受力的极限。机翼不仅需要支撑飞机在空中飞行，还需要在各种极端环境下维持其性能，如强风、气流扰动等。这就涉及机翼材料的强度极限、机翼结构的承载极限以及机翼的空气动力性能极限。

首先，机翼材料的强度极限指的是材料在力的作用下，能够抵抗变形或破裂的极限。这个极限是由材料的本质（如硬度、韧性等）决定的，而且在设计时必须考虑到。

其次，机翼结构的承载极限涉及飞机翼的设计和制造过程。机翼需要经受住飞行中的各种压力和扭矩，这就需要设计者精确计算和优化机翼的结构，确保其在极限状态下也能保持稳定。

最后，机翼的空气动力性能极限主要与飞机的飞行性能有关。机翼的形状和大小会影响其升力和阻力，进而影响飞机的飞行速度和耗油量。因此，机翼设计必须考虑到其在不同飞行状态（如起飞、巡航、降落等）下的空气动力性能极限。

机翼设计主要涉及的数学模型是流体力学中的纳维-斯托克斯（Navier-Stokes）方程，该方程描述了流体运动的基本规律。空气可以被视为一种流体，机翼在飞行中与空气的相互作用（如升力和阻力）可以通过纳维-斯托克斯方程来计算和分析。

简化的纳维-斯托克斯方程如下：

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g}$$

其中， ρ 是流体密度； \mathbf{u} 是流体速度； t 是时间； p 是压力； μ 是动力黏度； \mathbf{g} 是重力加速度。

机翼设计需要考虑的极限包括：

最大升力：在起飞和降落阶段，飞机需要尽可能大的升力。翼型设计、翼展和攻角都会影响升力。

最小阻力：在巡航阶段，飞机需要尽可能小的阻力，以提高燃料效率。阻力主要来自形状阻力和摩擦阻力，翼型设计和表面处理都会影响阻力。

最大速度：飞机的最大速度受到空气阻力和发动机推力的限制。当阻力等于推力时，飞机达到最大速度。

以上极限条件可以通过纳维-斯托克斯方程模拟得出。在设计过程中，工程师会通过改变翼型、翼展等参数模拟飞机在各种飞行状态下的动力性能，以满足以上的极限条件。

综上，飞机翼设计中的极限问题是一个复杂而重要的课题，需要设计者充分理解和运用极限的

概念，以确保飞机的安全和效率。

数值计算实践

可以使用一个简化的模型来帮助我们理解飞机翼的升力和阻力如何随速度变化。这种模型通常称为升阻曲线。

在这个模型中，飞机的升力系数和阻力系数会随着马赫数（飞机速度与声速之比）的增加而变化。当马赫数接近 1 时，升力会达到峰值，然后随着马赫数的进一步增加而下降；阻力则会在马赫数接近 1 时急剧增加，这是因为当飞机速度接近声速时会产生音爆，导致阻力大幅增加。扫描下方二维码可查看代码。

在这个 Python 示例代码中，使用 NumPy 和 Matplotlib 库来模拟和绘制马赫数序列下的升力系数和阻力系数。首先，生成了一个马赫数序列 M ，然后使用简化的升力系数模型和阻力系数模型计算了相应的升力系数（CL）和阻力系数（CD）。接着，通过绘制图形，将马赫数（ M ）作为横轴，升力系数（CL）和阻力系数（CD）分别作为纵轴，展示了这两个系数随马赫数变化的趋势，如图 1.18 所示。图形展示了随着马赫数的增加，升力系数逐渐减小，而阻力系数逐渐增大。

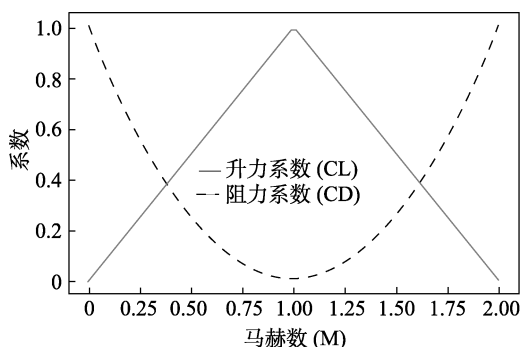


图 1.18 升力系数与阻力系数随马赫数的变化



示例代码

上面是一个简单的 Python 示例，模拟了这个过程并生成了相应的图表，这个模型非常简单，只是为了方便理解升力和阻力随速度的变化。实际的机翼设计需要考虑很多其他因素，如翼型、翼展、攻角等，需要使用更为复杂和精确的模型。

1.24 训练过程中的梯度消失问题应用和简化模型及数值计算实践

深度学习中的梯度消失就是在训练神经网络时，参数的梯度值变得非常小，以至于在反向传播时，权重更新的速度极慢，使得学习过程变得非常低效，甚至无法收敛。

梯度消失主要原因在于深度神经网络的训练中使用的激活函数（如 Sigmoid、tanh 等）和乘法运算。这些激活函数在输入值的绝对值较大时，其导数接近于 0，这会导致在反向传播时梯度值会连续地乘以小于 1 的数，从而变得非常小。神经网络的层数增多，梯度消失的问题会变得更加严重。使用 ReLU 函数、预训练与微调、使用残差网络（ResNet）、批量归一化（Batch Normalization）等，都能在一定程度上缓解或避免梯度消失的问题。

例如，ReLU 函数在输入值大于 0 时，其导数为 1，这避免了反向传播过程中梯度的持续衰减。预训练与微调的方法是先使用无监督的方式预训练浅层网络，再进行整体的微调，这种分步训练的方式也能在一定程度上缓解梯度消失的问题。残差网络则通过引入跳跃连接，直接将输入信号传递到后面的层，从而避免了反向传播过程中梯度的衰减。批量归一化是一种改善网络训练效果的技术，它通过对每一层的输入进行归一化处理，使得每一层的输入都服从标准正态分布，从而

在一定程度上缓解了梯度消失的问题。

ReLU 函数：ReLU 函数的表达式为 $f(x) = \max(0, x)$ 。它的导数在 $x > 0$ 时为 1，在 $x < 0$ 时为 0。这种特性使得 ReLU 函数在反向传播过程中，对于大于 0 的输入，梯度不会衰减。

预训练与微调：这种方法的数学表达比较复杂，一般不直接写出公式。简单来说，预训练是先在无监督任务（比如自编码器或预测下一个词）上训练神经网络的前几层，然后再在目标任务上进行微调。

残差网络：残差网络的关键是引入了跳跃连接（或称残差连接）。在数学上，如果一个普通的深度网络输出是 $y = F(x)$ ，那么引入残差连接后的网络输出就变成了 $y = x + F(x)$ ，这样在反向传播过程中至少有一部分梯度可以直接传递，避免了梯度消失的问题。

批量归一化：批量归一化的数学表达为 $\text{BN}(x) = \frac{\gamma(x - \mu)}{\sigma} + \beta$ ，其中 μ 和 σ 是输入 x 的均值和标准差， γ 和 β 是可学习的参数。通过这种方式，每一层的输入都被归一化为均值为 0、方差为 1 的分布，从而在一定程度上缓解了梯度消失的问题。

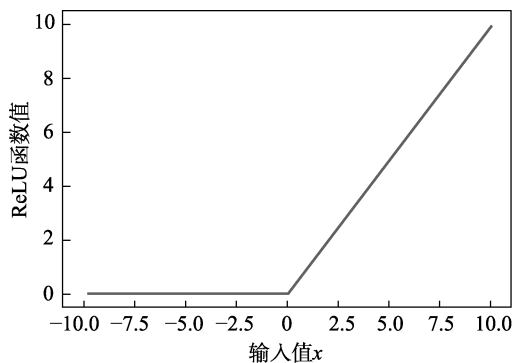
通过深度理解极限的概念和特性，可以更好地理解和解决深度学习中的梯度消失问题，进一步提升深度学习的训练效率和模型性能。

数值计算实践

我们模拟 ReLU 函数的工作原理并绘图，因为 ReLU 函数有一个明显的“极限”：当输入小于 0 时，函数值始终为 0。扫描下方二维码可查看代码。

在这个 Python 示例代码中，使用 NumPy 和 Matplotlib 库来绘制修正线性单元（ReLU）函数的图像。首先，创建一个 -10~10 的输入数据的序列 x 。然后，对输入数据应用 ReLU 函数，通过 `np.maximum(0, x)` 将所有负数变为零。最后，绘制图形，将输入数据 (x) 作为横轴，对应的 ReLU 输出 (y) 作为纵轴，展示 ReLU 函数的非线性特性：当输入为负数时，输出为 0；当输入为正数时，输出等于输入。

以下是一个简单 ReLU 函数的 Python 代码实现及其可视化展示。在图 1.19 中，ReLU 函数的输出在 x 小于 0 时一直为 0（这是一种形式的“极限”），而在 x 大于 0 时，输出与输入相同。



示例代码

图 1.19 ReLU 函数

1.25 习题、思考题、课程论文研究方向

► 习题：

1. 探索函数 $f(x) = \frac{\sin x}{x}$ 在 x 趋近于 0 时的极限值。
2. 计算极限 $\lim_{x \rightarrow \infty} \left(1 + \frac{1}{x}\right)^x$ ，并解释其与自然底数 e 的关系。

3. 研究函数 $f(x) = x^2 \sin\left(\frac{1}{x}\right)$ 在 x 趋近于 0 时的极限。

4. 判断函数 $g(x) = \frac{x-1}{x^3-1}$ 在 $x=1$ 处是否连续, 并解释原因。

思考题:

1. 如何证明极限 $f(x) = L$, $\lim_{x \rightarrow a} f(x) = L$ 的定义?

2. 探究极限与连续之间的关系, 解释为什么一个函数在某点处连续意味着在该点存在极限。

3. 考虑函数 $h(x) = \begin{cases} x^2, & x \leq 1 \\ 2x-1, & x > 1 \end{cases}$, 讨论 $h(x)$ 在 $x=1$ 处的极限与连续情况。

4. 分析函数 $f(x) = \begin{cases} \sin \frac{1}{x}, & x \neq 0 \\ 0, & x = 0 \end{cases}$ 的连续性和极限性质, 给出证明或反例。

课程论文研究方向:

1. 探究极限与连续的概念在实际应用中的数学模型和算法的开发。

2. 研究在科学、工程、经济等领域中使用极限与连续的数学模型的有效性和适用性。

3. 分析不同类型函数的极限和连续性, 并研究其在实际问题中的应用。

4. 深入探讨极限与连续的数学理论, 如实数系统、序列极限、函数极限等, 并与实际问题相联系。