第二章 模型估计与优化

在机器学习领域，很多机器学习模型的输入输出规则在本质上都可以看成是某种映射函数，作为初始模型的映射函数通常包含一组待定的未知参数，需要通过对训练样本的学习确定这些参数的合理取值。因此，机器学习中有一大类模型的求解过程实际上是解决这些未知参数的取值问题。通常使用对目标函数进行优化计算的方式获得参数取值。如果初始模型较为简单，则模型求解的目标函数通常也较为简单。对于目标函数为线性函数的情形，可用单纯形法等常用线性规划方法获得精确解，实现对所求优化模型的精确构造。然而，用于机器学习模型优化的目标函数主要是非线性函数或约束条件中含有非线性函数，通常称这类优化计算问题为非线性规划问题。目前还没有针对此类优化计算问题的通用精确解法，通常使用具有针对性的近似计算方法进行模型参数求解，构造具有一定精度的近似优化模型。对于比较简单的非线性目标函数，通常使用参数估计方式直接对模型参数进行近似估计。对于较为复杂的非线性目标函数，直接对其进行参数估计一般难以取得满意的效果，此时通常使用迭代计算或动态规划方式逐步优化模型参数估计值，使得模型性能得到逐步提升并达到最优或近似最优。此外，还需采用一些特定策略对模型做正则化处理尽量消除模型中可能存在的过拟合现象。本章主要介绍模型求解的近似计算方法，首先简要介绍模型参数估计的基本方法；然后介绍几种常用的模型优化近似计算方法，包括基本的近似优化方法和概率型近似优化方法；最后介绍模型正则化的基本概念和常用策略。

2.1 模型参数估计

对机器学习模型的参数直接进行估计是一种最简单最直观的模型求解思路。显然，机器学习模型的参数估计需要给出的是参数具体估计值，而不仅仅是参数的大致取值范围。因此，机器学习模型的参数估计方法均为点估计方法。对于给定的机器学习任务，同一种模型结构在采用不同模型参数时的性能一般会存在一定的差异，如何选择一组参数使得模型对具体任务的表现达到最优是参数估计要解决的关键问题。本节简要介绍最小二乘、最大似然和最大后验这三种机器学习中最常用的参数估计方法。

2.1.1 最小二乘估计

最小二乘估计是一种基于误差平方和最小化的参数估计方法。对于线性模型，其最小二乘估计量是一种具有最小方差的无偏估计量，由最小二乘法求得的参数估计值是最优估计值。此外，最小二乘法计算简单、易于理解且具有良好的实际意义。因此，最小二乘法是对线性统计模型进行参数估计的基本方法。

如前所述，对于任意一个给定的示例X，可将其表示为表征向量或特征向量的形式。不失一般性，将样本集合中每个示例分别看成是一个特征向量。假设训练样本集为：

$$\mathrm{S}=\{(X_1,y_1),(X_2,y_2),\ldots,(X_n,y_n)\}$$

可将其中示例X_i表示为特征向量$X_i=(x_{1i},x_{2i},\ldots,x_{ki})^T$，　x_{si}为示例X_i的第s个特征。

线性模型的初始模型一般可写成$f(X)=X^T\boldsymbol{\beta}$，其中$\boldsymbol{\beta}=(\beta_1,\beta_2,\ldots,\beta_k)^T$为待求的参数向量,$X$为某个示例的特征向量。对于训练样本集合中任意给定的一个示例X_i，模型参数$\boldsymbol{\beta}$的真实值应该尽可能使得模型对示例X_i的输出$f(X_i)$与该示例标注值y_i之间的误差达到最小。因此，从整体上看，如果存在参数向量的一组取值$\widehat{\boldsymbol{\beta}}$，线性模型能够在该组参数取值下获得模型输出与标注值之间在训练样本集上最小的整体误差，则将$\widehat{\boldsymbol{\beta}}$作为$\boldsymbol{\beta}$的估计值最为合理。

最小二乘法正是基于上述思想。用$f(X_i)-y_i$表示模型f对示例X_i的输出与该示例的真实值之间的误差。为防止误差正负值相互抵消和便于数学上的求导运算，最小二乘法将优化目标函数定义为样本个体误差的平方和，即有：

$$F(\boldsymbol{\beta})=\sum_{i=1}^{n}[f(X_i)-y_i]^2=\sum_{i=1}^{n}(X_i^T\boldsymbol{\beta}-y_i)^2$$

当目标函数取得最小值时，所对应模型参数为最优。由于函数极值点处对所有参数的偏导均为 0，故可由此求得最小二乘估计值。使用一个$n\times k$的矩阵$\boldsymbol{X}=(X_1,X_2,\ldots,X_n)^T$表示训练样本集，则线性模型可表示为$f(\boldsymbol{X})=\boldsymbol{X\beta}$，由此可得如下目标函数：

$$F(\boldsymbol{\beta})=(\mathbf{y}-\boldsymbol{X\beta})^T(\mathbf{y}-\boldsymbol{X\beta}) \qquad (2-1)$$

其中$F(\boldsymbol{\beta})$为向量形式的误差平方，$\mathbf{y}=(y_1,y_2,\ldots,y_n)^T$ 为训练样本集的标注值向量。

$F(\boldsymbol{\beta})$取得最小值时所对应参数向量$\widehat{\boldsymbol{\beta}}$即为最小二乘法的估计值，即有：

$$\widehat{\boldsymbol{\beta}}=arg_{\boldsymbol{\beta}}F(\boldsymbol{\beta})=\arg_{\boldsymbol{\beta}}\min(\mathbf{y}-\boldsymbol{X\beta})^T(\mathbf{y}-\boldsymbol{X\beta})$$

令$F(\boldsymbol{\beta})$对$\boldsymbol{\beta}$的偏导数为 0，可得方程组：$\boldsymbol{X}^T(\mathbf{y}-\boldsymbol{X\beta})=0$。解此方程组可得参数向量$\boldsymbol{\beta}$的最小二乘估计值为：

$$\widehat{\boldsymbol{\beta}}=(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\mathbf{y} \qquad (2-2)$$

【例题 2.1】已知某工厂产值Q与其劳动力投入L之间满足关系$Q=aL^b$，其中a,b为未知参数。试根据表 2-1 中数据确定劳动力投入L与工厂产值Q之间的关系。

表 2-1 劳动力投入与产值关系表

年份	2013	2014	2015	2016	2017
劳动力投入L（万元）	42	51	49	65	57
产值Q（万元）	188	210	194	207	221

【解】工厂产值Q与其劳动力投入L和资金投入K之间并非满足线性关系，但可在等式两边同时取对数将其转化为线性关系：$\ln Q=\ln a+b\ln L$。令：

$$y=\ln Q,\ x=\ln L;\ \beta_0=\ln a,\ \beta_1=b$$

将示例X_i定义为一个包含两个元素的列向量，其中第一个元素恒为 1，第二个元素为$x=\ln L$，即$X_i=(1,x_i)^T$,则可将原方程转化为线性统计模型$f(X)=\boldsymbol{\beta}X$，其中$\boldsymbol{\beta}=(\beta_0,\beta_1)$是为参数向量。依据最小二乘估计方法构造优化目标如下：

$$F(\boldsymbol{\beta})=\sum_{i=1}^{5}[f(X_i)-y_i]^2=\sum_{i=1}^{5}(\beta_0+\beta_1x_i-y_i)^2$$

将目标函数$F(\boldsymbol{\beta})$分别对参数向量中元素β_0,β_1求偏导并令导数值为 0，有：

$$\frac{\partial F}{\partial\beta_0}=2\sum_{i=1}^{n}(\beta_0+\beta_1x_i-y_i)(-1)=0$$

$$\frac{\partial F}{\partial \beta_1} = 2\sum_{i=1}^{n}(\beta_0+\beta_1 x_i - y_i)(-X_i) = 0$$

代入数据算得$\widehat{\beta_0} = 4.1952$，$\widehat{\beta_1} = 0.2835$。故有$a = e^{4.1952} \approx 66.37$，$b = 0.2835$。由此得到该工厂产值$Q$与其劳动力投入$L$之间满足数量关系：$Q = 66.37L^{0.2835}$。□

2.1.2 最大似然估计

在机器学习领域，为了能够有效计算和表达样本出现的概率，通常假定面向同一任务的样本服从相同的带有某种或某些参数的概率分布。如果能够求出样本概率分布的所有未知参数，则可使用该分布对所有样本进行分析。最大似然估计是一种基于概率最大化的概率分布参数估计方法。该方法将当前已出现样本类型看作一个已发生事件。既然该事件已经出现，就可假设其出现的概率最大。因此，样本概率分布的参数估计值应使得该事件出现的概率最大。这就是最大似然估计方法的基本思想。

假设样本X为离散随机变量，其概率分布函数为$p(X;\boldsymbol{\beta})$，即有$p(X_i|\boldsymbol{\beta}) = P(X = X_i)$。其中$\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_k)^T$为未知参数向量。假设从样本总体中随机抽取$n$个样本$X_1, X_2, \ldots, X_n$，则可将“从总体中随机抽取到$X_1, X_2, \ldots, X_n$这$n$个样本”记为一个事件$A$。事件$A$发生的概率可用下列函数度量：

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} p(X_i|\boldsymbol{\beta}) \qquad (2-3)$$

上述函数是一个关于未知参数向量$\boldsymbol{\beta}$的函数，通常称之为**似然函数**。既然事件A已经发生，那么该事件发生的概率应该最大。故可将未知参数向量$\boldsymbol{\beta}$的估计问题转化为求似然函数$L(\boldsymbol{\beta})$最大值的优化问题，即最大似然估计值为：

$$\widehat{\boldsymbol{\beta}} = \arg_{\boldsymbol{\beta}} \max L(\boldsymbol{\beta}) = \arg_{\boldsymbol{\beta}} \max \prod_{i=1}^{n} p(x_i|\boldsymbol{\beta})$$

【例 2.2】假设一个不透明盒里装有 3 颗围棋子，现用有放回抽样法随机抽取三次，每次拿一颗，得到白子 2 次黑子 1 次。试用最大似然估计法估计盒中白子个数。

【解】设盒中有$\theta(\theta = 0,1,2,3)$枚白子，$p(白|\theta)$为在一次采样中抽到白子的概率分布，则有：

当$\theta = 0$时，$p(白|\theta) = 0$；当$\theta = 1$时，$p(白|\theta) = 1/3$；

当$\theta = 2$时，$p(白|\theta) = 2/3$；当$\theta = 3$时，$p(白|\theta) = 1$。

由于三次采样中抽到了两次白子，故似然函数为$L(\theta) = [p(白|\theta)]^2[1 - p(白|\theta)]$。分别取$\theta = 0,1,2,3$，可得$L(0) = 0$，$L(1) = 2/9$，$L(2) = 4/9$，$L(3) = 0$。为使得事件“三次采样抽中两次白子”发生概率最大，应取$\hat{\theta} = 2$作为参数θ的最大似然估计，此时似然函数取最大值$4/9$。□

当样本X为连续随机变量时，可用其概率密度函$f(X;\boldsymbol{\beta})$构造似然函数$L(\boldsymbol{\beta})$，即有：

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} f(X_i;\boldsymbol{\beta}) \qquad (2-4)$$

对似然函数$L(\boldsymbol{\beta})$作最大优化计算即可得到对参数$\boldsymbol{\beta}$的估计值，即$\widehat{\boldsymbol{\beta}} = \arg_{\boldsymbol{\beta}} \max L(\boldsymbol{\beta})$。由于$L(\boldsymbol{\beta})$为多个函数连乘，难以求解，故取自然对数运算将其转化为累加形式的对数似然函数$\ln L(\boldsymbol{\beta})$。自然对数函数为严格单调递增函数， $L(\boldsymbol{\beta})$与$\ln L(\boldsymbol{\beta})$具有相同的极值点，故$L(\boldsymbol{\beta})$与$\ln L(\boldsymbol{\beta})$具有相同的优化效果。对数似然函数$\ln L(\boldsymbol{\beta})$的具体形式为：

$$\ln L(\boldsymbol{\beta}) = \sum_{i=1}^{n} ln\, f(X_i; \boldsymbol{\beta}) \qquad (2\text{-}5)$$

可通过对数似然$\ln L(\boldsymbol{\beta})$的优化计算获得似然函数$L(\boldsymbol{\beta})$的最优解，即有：

$$\widehat{\boldsymbol{\beta}} = \arg_{\boldsymbol{\beta}} \max \ln L(\boldsymbol{\beta}) = \arg_{\boldsymbol{\beta}} \max L(\boldsymbol{\beta})。$$

【例题 2.2】已知某校学生身高服从高斯分布$N(\mu, \sigma^2)$,现从全体学生中随机抽取 10 位同学，测得他们身高如表 2-2 所示。试根据表中数据估计该校学生身高的均值和方差。

表 2-2 学生身高表

编号k	1	2	3	4	5	6	7	8	9	10
身高X(cm)	171	164	174	165	168	181	176	162	173	172

【解】已知高斯分布的概率密度函数为：

$$f(X_k; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp[-\frac{(X-\mu)^2}{2\sigma^2}]$$

其中μ和σ^2分别为方差，X_k表示 k号的学生身高。由此可得如下似然函数：

$$L(\mu, \sigma^2) = \prod_{k=1}^{10} f(X_k; \mu, \sigma^2) = \prod_{k=1}^{10} \frac{1}{\sqrt{2\pi}\sigma} \exp[-\frac{(X_k-\mu)^2}{2\sigma^2}]$$

对数似然为：

$$\ln L(\mu, \sigma^2) = -5 \ln 2\pi - 10 \ln \sigma - \frac{\sum_{k=1}^{10}(X_k-\mu)^2}{2\sigma^2}$$

对$\ln L(\mu, \sigma^2)$分别求μ和σ^2的偏导并令导数值为 0，可得：

$$\frac{\partial \ln L(\mu, \sigma^2)}{\partial \mu} = \frac{\sum_{k=1}^{10}(X_k-\mu)}{\sigma^2} = 0$$

$$\frac{\partial \ln L(\mu, \sigma^2)}{\partial \sigma^2} = -\frac{5}{\sigma^2} + \frac{\sum_{k=1}^{10}(X_k-\mu)^2}{2\sigma^4} = 0$$

解得：

$$\hat{\mu} = \bar{X} = \frac{1}{10}\sum_{k=1}^{10} X_k \quad ; \quad \widehat{\sigma^2} = \frac{1}{10}\sum_{k=1}^{10}(X_k - \bar{X})^2$$

代入数据可算得学生身高均值和方差的最大似然估计分别为$\hat{\mu} = 170.6$，$\widehat{\sigma^2} = 31.24$。□

2.1.3 最大后验估计

最大后验估计是一种结合过往经验的参数估计方法。与最大似然估计认为待求参数是某个固定未知取值不同，最大后验估计认为待求参数服从某一未知概率分布，参数以一定的概率取某一特定值。在进行参数估计时，最大后验估计依据过往经验和已经出现的样本共同确定参数的可能取值。以抛掷硬币实验为例，现在希望估计硬币正面向上的概率θ，依据过往

经验，硬币正面向上概率θ一般为0.5，但考虑到硬币个体可能会存在某些特点，故没有将θ值确定为0.5，而是给出关于θ取值的一个概率分布函数$g(\theta)$，比如令：

$$g(\theta)=\begin{cases}0.9, & \theta=0.5\\ 0.1, & \theta\neq 0.5\end{cases}$$

$g(\theta)$被称为对参数θ的**先验概率分布**或**先验概率**，表示根据过往经验得到θ取值的概率。假如抛掷完成 10 次硬币，其中 7 次正面向上，3 次反面向上，则最大后验估计希望根据样本出现情况对参数取值进行估计，即考虑在样本取值已经出现的情况下计算θ取值的条件概率$f(\theta|X)$,其中X表示已经出现的样本取值情况，$f(\theta|X)$被称为**后验概率**，可看成是根据样本数据出现的实际情况对先验概率$g(\theta)$的某种修正。后验概率最大时所对应参数取值即为所求的最大后验估计值，即有：

$$\hat{\theta}=\arg_{\theta}\max f(\theta|X) \qquad (2-6)$$

由贝叶斯公式可知后验概率$f(\theta|X)$的计算公式如下：

$$f(\theta|\mathrm{X})=\frac{f(X|\theta)g(\theta)}{p(X)} \qquad (2-7)$$

其中$f(X|\theta)$为现有样本所表现出的信息，分母$p(\mathrm{X})$为样本分布。

显然，$p(\mathrm{X})$与参数θ无关且恒大于零，故可直接通过最大化$f(X|\theta)g(\theta)$的优化方式实现最大后验估计，即有：

$$\hat{\theta}=\arg_{\theta}\max f(X|\theta)g(\theta) \qquad (2-8)$$

由以上分析可知，最大后验估计通过综合考虑参数θ的先验信息$g(\theta)$和现有样本信息$f(X|\theta)$确定参数的估计值。

继续讨论对上述抛掷硬币实验的概率估计问题，由于$g(\theta=0.5)=0.9$，故在$\theta=0.5$的条件下抛掷十次硬币发生事件“7 次正面向上，3 次反面向上”的概率为：

$$f(X=7{,}3|\theta=0.5)=C_{10}^{7}\theta^{7}(1-\theta)^{3}=0.1171875$$

其中“$X=7{,}3$”表示抛掷十次硬币发生事件“7 次正面向上，3 次反面向上”。

由此可得：

$$f(X=7{,}3|\theta=0.5)g(\theta=0.5)=0.10546875$$

由于$f(X=7{,}3|\theta\neq 0.5)$是一个概率值，故有$f(X=7{,}3|\theta\neq 0.5)\leq 1$，从而有：

$$f(X=7{,}3|\theta\neq 0.5)g(\theta\neq 0.5)\leq 0.1<f(X=7{,}3|\theta=0.5)g(\theta=0.5)$$

根据最大后验估计理论可知：

$$\hat{\theta}=\arg_{\theta}\max f(\mathrm{X}=7{,}3|\theta=0.5)g(\theta=0.5)=0.5$$

即硬币正面向上概率的最大后验估计值$\hat{\theta}=0.5$。

由上述分析可知，尽管已知样本的取值状况与过往经验不相符，但由于过往经验较为可靠，故最大后验估计在结论上选择相信了经验而非实际样本所表现出的信息，即认为已知样本取值状况与过往经验不相符的原因是由随机波动造成。若使用最大似然估计方法对上述情况进行参数估计，则得到估计值为$\hat{\theta}=0.7$。但由于实验次数较少，实验结果可能存在较大波动。因此，如果在这种情况下使用只考虑样本信息的最大似然方法，则所得到的估计值可能会与参数的真实值存在较大差异。

一般地，在对多个未知参数进行估计时，可将最大后验估计表示为：

$$\hat{\boldsymbol{\beta}}=\arg_{\boldsymbol{\beta}}\max f(\mathrm{X}|\boldsymbol{\beta})g(\boldsymbol{\beta}) \qquad (2-9)$$

其中$\boldsymbol{\beta}=(\beta_1,\beta_2,\ldots,\beta_k)^T$为未知参数向量。

亦可将(2-9)式所示的目标函数取自然对数，得到与之等价的对数形式：

$$\widehat{\boldsymbol{\beta}}=\arg_{\boldsymbol{\beta}}\max(\ln f(\mathrm{X}|\boldsymbol{\beta})+\ln g(\boldsymbol{\beta})) \qquad (2-10)$$

【例题 2.3】假设某公司员工收入过去三年均服从均值为 6（万元），方差为 0.36（万元）的高斯分布，表 2-3 表示从公司随机抽取 10 名员工的收入数据，试根据表中数据和过去员工的收入情况估计今年员工收入的均值和方差。

表 2-3 某公司员工年收入数据

编号k	1	2	3	4	5	6	7	8	9	10
收入X(万元)	6.1	5.3	7.1	7.3	6.4	5.9	6.7	6.3	5.6	6.5

【解】已知高斯分布的概率密度函数为：

$$f(X;\mu,\sigma^2)=\frac{1}{\sqrt{2\pi}\sigma}\exp[-\frac{(X-\mu)^2}{2\sigma^2}]$$

依题意可知，收入X的先验概率为:

$$f(X;6,0.36)=\frac{1}{\sqrt{2\pi}\times 0.6}\exp[-\frac{(X-6)^2}{0.72}]$$

后验概率为：

$$f(\mu,\sigma^2|X_k)=f(X;6,0.36)]\prod_{k=1}^{10}f(X_k|\mu,\sigma^2)$$

$$=\frac{1}{\sqrt{2\pi}\times 0.6}\exp[-\frac{(\mu-6)^2}{0.72}]\prod_{k=1}^{10}\frac{1}{\sqrt{2\pi}\sigma}\exp[-\frac{(X_k-\mu)^2}{2\sigma^2}]$$

为求最大后验估计值，对上式取对数后分别对μ和σ^2求偏导并令导数值为 0：

$$\ln f(\mu,\sigma^2|X_k)==-\ln\sqrt{0.72\pi}-\frac{(\mu-6)^2}{0.72}-5\ln 2\pi\sigma-\frac{\sum_{k=1}^{10}(X_k-\mu)^2}{2\sigma^2}$$

$$\frac{\partial\ln f(\mu,\sigma^2|X_k)}{\partial\mu}=-\frac{\mu-6}{0.36}+\frac{\sum_{k=1}^{10}(X_k-\mu)}{\sigma^2}=0$$

$$\frac{\partial\ln f(\mu,\sigma^2|X_k)}{\partial\sigma^2}=-\frac{5}{2\sigma^2}+\frac{\sum_{k=1}^{10}(X_k-\mu)^2}{2\sigma^4}=0$$

解得：

$$\hat{\mu}=\frac{10\times 0.36}{10\times 0.36+\hat{\sigma}^2}\left(\frac{1}{10}\sum_{k=1}^{10}X_k\right)+\frac{6\hat{\sigma}^2}{10\times 0.36+\hat{\sigma}^2}$$

$$\hat{\sigma}^2=\frac{\sum_{k=1}^{10}(\hat{\mu}-X_k)^2}{5}$$

将上面两式进行联立并将表 2-3 中数据代入，解得今年员工收入均值和方差的最大后验估计值为：$\hat{\mu}=6.4$；$\hat{\sigma}^2=0.72$。□

2.2 模型优化基本方法

在优化目标较为复杂时，通常很难直接通过参数估计方法求得最优估计值。事实上，机

器学习的模型训练除了使用前述参数估计法之外，还可通过数值优化计算方法确定模型参数。这类数值优化方法通常采用迭代逼近方式确定最优解。在逼近最优解过程中，模型性能会逐渐提升，故称此类方法为模型优化方法。由于模型优化方法采用迭代方式逼近最优解的策略，故在很多情况下能够有效应对优化目标较为复杂的情况。机器学习的模型优化方法有很多，本节主要介绍两种基本方法，即梯度下降方法和牛顿迭代法。

2.2.1 梯度下降法

梯度下降方法是机器学习最常用的模型优化方法之一，其基本思想是朝着函数梯度的反方向不断迭代更新参数。由于梯度方向为函数值上升最快的方向，故梯度反方向就是函数值下降最快的方向。一直朝着梯度反方向更新参数可以使得函数值得到最快地下降，从而能够尽可能快速地逼近函数极小值点直至收敛。梯度下降方法的数学表达如下：

$$X_{k+1} = X_k + step_k P_k \qquad (2-11)$$

其中$step_k$为第k次迭代的步长，P_k为第k次寻优方向，即为梯度反方向$P_k = -\nabla F(X_k)$。

$(2-11)$式的含义是在第k次迭代起始点X_k确定的情况下，向目标函数梯度反方向走一段距离并将此次所到新位置$X_k + step_k P_k$作为下次迭代起点赋值给X_{k+1}。通过对$step_k$适当取值就可由此得到目标函数的最优解。图 2-1 表示初始迭代点为X_1的梯度下降迭代过程。

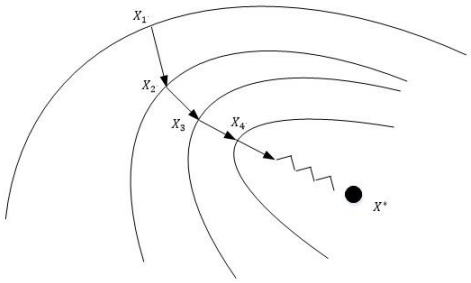


图 2-1 梯度下降方法的迭代过程

梯度下降方法的关键在于如何确定每次迭代的搜索方向和迭代步长。以图 2-1 所示的迭代过程为例，从起始点X_1开始通过梯度下降法进行迭代优化，则有：

$$X_2 = X_1 + step_1 P_1$$

其中$P_1 = -\nabla F(X_1)$，

令$F(X)$为优化的目标函数，则步长$step_1$可通过下列优化方式确定：

$$\text{argmin}_{step_1 \geq 0} F(X_1 + step_1 P_1) \qquad (2-12)$$

现给出步长$step_k$的具体计算公式，根据二次泰勒展开式可将目标函数$F(X)$其近似表示为正定二次函数：

$$F(X) = \frac{1}{2} X^T \boldsymbol{A} X + b^T X + c \qquad (2-13)$$

其中$\boldsymbol{A}$为正定的系数矩阵，X为参数向量，b为常数向量，c为常数。

在X_k处对$F(X)$求梯度可得$P_k = \nabla F(X_k) = \boldsymbol{A} X_k + b^T$。从$X_k$点出发沿着梯度的反方向进行搜索，则有：

$$X_{k+1} = X_k - step_k \nabla F(X_k) \qquad (2-14)$$

当选择最优步长时，每步搜索方向均与上步搜索方向正交，即有$P_{k+1}^T P_k = 0$。将P_{k+1}^T展

开，则有$[\boldsymbol{A}(X_k - step_k P_k) + b^T]P_k = 0$，由此解出$step_k$并将其代入迭代公式(2-14)，则可将梯度下降迭代公式进一步改写为：

$$X_{k+1} = X_k - \frac{{P_k}^T P_k}{{P_k}^T \boldsymbol{A} P_k} P_k \qquad (2-15)$$

在机器学习的具体应用中，梯度下降方法的步长有时会根据需要通过人为设定，这需要一定的经验。如果步长设定过大，则会导致算法不收敛；如果步长设定过小，则会使得算法收敛较慢，提高计算的时间成本。

例如，对于函数问题$\min F(X) = x_1^2 + 4x_2^2$，显然有：

$$\boldsymbol{A} = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix}$$

假设起始点为$X_1 = [1,1]^T$，则有$F(X_1) = 5$，$P_1 = -\nabla F(X_1) = [2,8]^T$。由迭代公式可得：

$$X_2 = [1,1]^T - 0.13077[2,8]^T = [0.73846, -0.04616]^T$$

$$F(X_2) = 0.06134$$

$$P_2 = -\nabla F(X_2) = [0.22152, 0.88008]^T$$

若迭代次数允许，则可一直迭代下去，直到满足终止条件，得到近似最优解。

【例题 2.4】试根据表 2-4 中数据建立线性回归模型，并使用该模型预测面积为 137m^2 的房屋价格，要求其中对目标函数的优化采用梯度下降法。

表 2-4 房屋价格与房屋尺寸数据

序号	1	2	3	4	5	6	7	8	9	10
面积S（m^2）	110	140	142.5	155	160	170	177	187.5	235	245
价格P(万元)	199	245	319	240	312	279	310	308	405	324

【解】表 2-4 中数据较大，不方便计算，因此此处先对其进行归一化处理再求解线性回归模型，具体方式为：

$$X = (S_i - S_{\min})/(S_{\max} - S_{\min})；\ y = (P_i - P_{\min})/(P_{\max} - P_{\min})$$

其中$S_{\min}$和$S_{\max}$分别表示最小和最大的房屋尺寸取值，S_i表示序号为i的房屋的尺寸取值，$P_{\min}$和$P_{\max}$分别表示最小和最大的房屋价格取值，P_i表示序号为i的房屋的价格取值。

经过归一化后的数据如表 2-5 所示。

表 2-5 归一化后的数据

序号	1	2	3	4	5	6	7	8	9	10
X	0.00	0.22	0.24	0.33	0.37	0.44	0.44	0.57	0.93	1.00
y	0.00	0.22	0.58	0.20	0.55	0.39	0.54	0.53	1.00	0.61

假设模型的具体形式为$y = w_1 X + w_0$。使用该模型构建目标函数，并采用误差平方和作为优化目标。为方便计算，将目标函数定义为$1/2$倍的误差平方和，即：

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{n}\left(y^i - y_p^i\right)^2 = \frac{1}{2}\sum_{i=1}^{n}\left(y^i - w_1 X^i - w_0\right)^2$$

其中y^i为序号为i的数据的真实值，y_p^i为对应的预测值，$\boldsymbol{w} = (w_0, w_1)^{\boldsymbol{T}}$为参数向量。

使用梯度下降方法对上述目标函数进行优化，通过如下迭代公式更新参数向量：

$$\boldsymbol{w}_{\text{new}} = \boldsymbol{w}_{\text{old}} - \eta \nabla E(\boldsymbol{w})$$

其中η为步长，此处步长选定为$\eta = 0.01$，$\boldsymbol{w}_{\text{old}}$表示当前更新的起点，$\boldsymbol{w}_{\text{new}}$表示更新后的权重向量。目标函数$E(\boldsymbol{w})$的梯度为：

$$\nabla E(\boldsymbol{w}) = -\sum_{i=1}^{n}(y^i - y_p^i)X^i$$

由此可将梯度下降算法的迭代计算公式转化为：

$$w_{\text{new}} = w_{\text{old}} + \eta\sum_{i=1}^{n}(y^i - y_p^i)X^i$$

设置$\boldsymbol{w}^0 = (1,1)^T$对上式进行 1000 次迭代，通过$Python$编程计算可得如表 2-6 所示的计算结果，表中仅给出部分迭代结果。

表 2-6 梯度下降方法迭代取值表

t(迭代次数)	w_1	w_0	$\nabla E(w_1)$	$\nabla E(w_0)$
1	0.952396	0.900800	4.760400	9.920000
2	0.910690	0.813681	4.170609	8.711878
3	0.874160	0.737168	3.652941	7.651344
…	…	…	…	…
999	0.704036	0.142370	-0.000058	0.000028
1000	0.704037	0.142370	-0.000057	0.000028

由表 2-6 中数据可知，经过 1000 次迭代后算法趋于收敛。因此可根据梯度下降方法求得线性回归模型为$y = 0.704037X + 0.142370$。对面积为 137m^2的房屋价格进行预测时，应先对该面积数据进行归一化计算，得到归一化后数据为$X = 0.2$。将其代入回归模型计算的对应的预测输出为$y = 0.2831774$，即得房屋价格预测值为 257.33 万元。□

梯度下降法在靠近极小值时收敛速度通常会减慢，使得计算效率下降。人们为此提出很多改进策略，共轭梯度下降法就是其中之一。共轭梯度下降法最初为求解非线性方程组而提出，后被推广到求解无约束优化问题，并逐渐成为最具代表性的最优化方法之一。该算法思想与梯度下降方法的相同之处在于都有着沿目标函数负梯度方向搜索的步骤；不同点在于梯度下降方法的搜索方向一直是负梯度方向，共轭梯度下降法的搜索方向从第二次确定搜索方向时，不再采用负梯度方向，而是经修正后的方向。因此，如何修正下次迭代的搜索方向是共轭梯度下降法的关键技术。下面具体介绍共轭梯度下降法。首先给出共轭的概念：

设$\boldsymbol{A}$为$R^{n\times n}$上对称正定矩阵，Q_1, Q_2为R^n上两个非零向量，若有${Q_1}^T\boldsymbol{A}Q_2 = 0$，则称$Q_1$与$Q_2$关于矩阵$\boldsymbol{A}$**共轭**，向量$Q_1$与$Q_2$的方向为一组**共轭方向**。

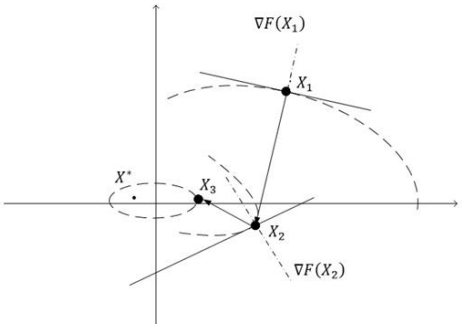


图 2-2 共轭梯度下降法

共轭梯度下降法的基本思路如图 2-2 所示。首先，任意选取初始点X_1，计算目标函数在该点梯度值$\nabla F(X_1)$，并将负梯度方向作为初次搜索方向，即$P_1=-\nabla F(X_1)$；然后，按图中箭头方向搜索下一点，即按公式$X_{k+1}=X_k+\alpha_k P_k$计算下一点X_2，其中α_k表示第k次迭代步长，为$\text{argmin}_{\alpha\geq 0}F(X_k+\alpha_k P_k)$的优化值。

搜索到X_2后，计算该点对应的梯度值$\nabla F(X_2)$，并按下式调整搜索方向：

$$P_{k+1}=-\nabla F(X_{k+1})+step_k P_k \qquad (2-16)$$

其中$step_k$为调整搜索方向时的步长。将$(2-16)$式两侧同时乘以$\boldsymbol{A}P_k$可得：

$$P_{k+1}^T\boldsymbol{A}P_k=-\nabla F(X_{k+1})\boldsymbol{A}P_k+step_k P_k^T\boldsymbol{A}P_k \qquad (2-17)$$

将步长$step_k$调整为$step_{k+1}$，使得P_{k+1}和P_k关于$\boldsymbol{A}$共轭，即有$P_{k+1}^T\boldsymbol{A}P_k=0$，可得：

$$step_{k+1}=\frac{\nabla F(X_{k+1})\boldsymbol{A}P_k}{{P_k}^T\boldsymbol{A}P_k} \qquad (2-18)$$

重复以上步骤，即可得到逼近最优解的序列$\{X_1,X_2,\dots,X_n,\dots\}$。

例如，对于优化问题$\min F(X)=2x_1^2+x_2^2$，取$X_1=[2,2]^T$为迭代初始值，由可得初次的搜索方向$P_1=-\nabla F(X_1)=-[8,4]^T$，并按下式计算$X_2$：

$$X_2=X_1+\alpha_1 P_1=[2-8\alpha_1,2-4\alpha_1]^T$$

首先通过优化问题$\text{argmin}[2(2-8\alpha_1)^2+(2-4\alpha_1)^2]$求出$\alpha_1=5/18$，然后由此算出$X_2=(-2/9,8/9)^T$和$P_2=-\nabla F(X_2)+step_1 P_1=(-8/9,16/9)^T$。再由$(2-18)$式算出$\alpha_2=9/20$，由此算出函数极值点$X_3=(0,0)^T$。

表 2-7 **UCI_IRIS**数据训练集中部分数据

编号	花萼长度	花萼宽度	花瓣长度	花瓣宽度	种类
1	6.4	2.8	5.6	2.2	2
2	5	2.3	3.3	1	1
3	4.9	2.5	4.5	1.7	2
4	4.9	3.1	1.5	0.1	0
5	5.7	3.8	1.7	0.3	0
6	4.4	3.2	1.3	0.2	0
7	5.4	3.4	1.5	0.4	0
…	…	…	…	…	…
119	4.4	2.9	1.4	0.2	0
120	4.8	3	1.4	0.1	0
121	5.5	2.4	3.7	1	1

【例题 2.5】UCI_IRIS数据集是一个常用训练数据集，共有 121 条数据，表 2-7 为其中部分数据。试用UCI_IRIS数据集和共轭梯度下降法训练一个多层神经网络模型。

【解】由表 2-7 可知，UCI_IRIS数据集中每个示例包含 4 个特征，所有示例分属 3 个类别。故感知机模型输入层应包含 4 个特征输入节点及 1 个偏置输入节点，输出层应包含 3 个输出节点。由此可构建如图 2-3 所示具有 10 个隐含节点的神经网络模型。

令w_{ki}^l为第l层第i个神经元与第$l+1$层第k个神经元之间的连接权重为w_{ki}^l，b_i^l为第l层第i个节点的偏置项，第l层激活函数表示为φ^l，则对于样本输入$X=(x_1,x_2,x_3,x_4)^T$，该模型第j个隐层节点的输出h_j为：

$$h_j = \varphi^2\left(\sum_{i=1}^{n} w_{ji}^1 x_i + b_j^2\right)$$

将上式表示为矩阵形式, 则有：

$$\boldsymbol{h} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{10} \end{bmatrix} = \varphi^2\left[\begin{pmatrix} w_{11}^1 & w_{12}^1 & w_{13}^1 & w_{14}^1 \\ w_{21}^1 & w_{22}^1 & w_{23}^1 & w_{24}^1 \\ \vdots & \vdots & \vdots & \vdots \\ w_{101}^1 & w_{102}^1 & w_{103}^1 & w_{104}^1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} b_1^2 \\ b_2^2 \\ \vdots \\ b_{10}^3 \end{pmatrix}\right] = \varphi^2(\boldsymbol{w}^1 X + \boldsymbol{b}^2)$$

同理可得该模型的输出$f(X)$为：

$$\boldsymbol{f}(X) = \begin{bmatrix} f_1(X) \\ f_2(X) \\ f_3(X) \end{bmatrix} = \varphi^3\left[\begin{pmatrix} w_{11}^2 & w_{12}^2 & \dots & w_{110}^2 \\ w_{21}^2 & w_{22}^2 & \dots & w_{210}^2 \\ w_{31}^2 & w_{32}^2 & \dots & w_{310}^2 \end{pmatrix}\begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_{10} \end{pmatrix} + \begin{pmatrix} b_1^3 \\ b_2^3 \\ b_3^3 \end{pmatrix}\right] = \varphi^3(\boldsymbol{w}^2 \boldsymbol{h} + \boldsymbol{b}^3)$$

其中φ^2为Sigmoid激活函数，φ^3为softmax激活函数，该激活函数可将模型输出映射为伪概率形式。

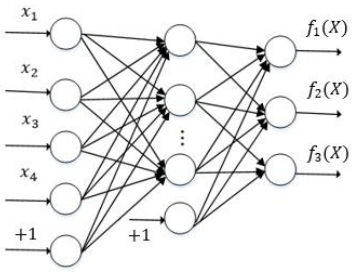


图 2-3 包含 10 隐含个节点的多层神经网络

通过对目标函数的优化计算方式估计模型参数。可将目标函数定义为模型输出在训练集上的平均误差，通过该误差(目标函数)的最小化实现对模型的训练构造。具体地说，使用如下(2-19)式作为目标函数，该目标函数依据模型对样本输出类别的概率对错分样本施加一定惩罚，并将对所有错分样本所加惩罚的均值作为模型输出在训练集上平均误差。

$$R(f) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{3} y_{ij} \log f_j(X_i) \qquad (2-19)$$

其中$f_j(X_i)$表示模型第j个输出节点对样本X_i的输出，y_{ij}为样本X_i所对应的标签向量$\boldsymbol{y}_i$中第j个元素的取值。

代入数据并用共轭梯度算法优化上述目标函数，通过Tersonflow框架编程计算可得权重更新结果，表 2-8 为输入层到隐藏层部分连接权重的部分计算数据，表 2-9 为隐藏层到输出层的部分连接权重的部分计算数据，取值保留小数点后两位。

表 2-8 输入层到隐藏层的部分连接权重取值

次数	w_{11}^1	w_{21}^1	…	w_{12}^1	w_{22}^1	…	w_{13}^1	w_{23}^1	…	w_{14}^1	w_{24}^1	…	b^1
0	-0.81	1.48	…	-2.44	0.09	…	0.59	-2.12	…	-0.04	0.89	…	-0.79
1000	-0.89	0.33	…	0.58	-1.33	…	-1.76	0.44	…	-0.04	1.56	…	-1.94
2000	-0.89	0.21	…	0.58	-1.47	…	-1.76	0.65	…	-0.04	1.83	…	-2.61
3000	-0.89	0.27	…	0.58	-1.52	…	-1.76	0.83	…	-0.04	2.05	…	-3.11

…	…	…	…	…	…	…	…	…	…	…	…	…	…
100000	-0.90	-0.66	…	0.58	3.26	…	-1.77	-3.37	…	-0.04	18.64	…	-20.0

表 2-9 隐藏层到输出层的部分连接权重取值

次数	w_{11}^2	w_{21}^2	…	w_{12}^2	w_{22}^2	…	w_{13}^2	w_{23}^2	…	b^2
0	-0.81	-2.44	…	1.48	0.09	…	0.06	0.59	…	-0.80
1000	-0.80	-3.46	…	1.47	-0.02	…	0.06	1.73	…	-0.81
2000	-0.80	-3.78	…	1.47	-0.09	…	0.06	2.13	…	-0.81
3000	-0.80	-3.97	…	1.47	-0.20	…	0.06	2.43	…	-0.81
…	…	…	…	…	…	…	…	…	…	…
100000	-0.80	-5.30	…	1.48	-8.70	…	0.06	12.25	…	-0.81

取满足精度要求的第 100000 迭代得到的连接权重$\boldsymbol{w}^{*1}, \boldsymbol{w}^{*2}$和偏置$\boldsymbol{b}^{*1}, \boldsymbol{b}^{*2}$作为最终模型参数，由此得到所求的分类映射规则：

$$f(\boldsymbol{X}) = \varphi[\boldsymbol{w}^{*2}\varphi(\boldsymbol{w}^{*1}\boldsymbol{x} + \boldsymbol{b}^{*1}) + \boldsymbol{b}^{*2}]$$

使用所求模型对如表 2-10 所示的测试数据进行预测，得到表中最后一列的预测计算结果。与该表中所示实际种类值进行比较，可知预测结果均为正确。□

表 2-10 测试数据与计算结果比较

编号	花萼长度	花萼宽度	花瓣长度	花瓣宽度	种类	预测值
1	5.9	3	4.2	1.5	1	1
2	6.9	3.1	5.4	2.1	2	2
3	5.1	3.3	1.7	0.5	0	0
4	6	3.4	4.5	1.6	1	1
5	5.5	2.5	4	1.3	1	1
6	6.2	2.9	4.3	1.3	1	1
7	5.5	4.2	1.4	0.2	0	0
8	6.3	2.8	5.1	1.5	2	2
9	5.6	3	4.1	1.3	1	1
10	6.7	2.5	5.8	1.8	2	2

共轭梯度下降法可以看成是梯度下降法的一种改进策略，仅需一阶导数信息，并克服了梯度法迭代后期收敛速度较慢的不足，是一种比较有效的优化算法。

2.2.2 牛顿迭代法

牛顿法是一种快速迭代搜索算法，主要用于求函数零点，即求方程的根。该算法要求目标函数具有二阶连续偏导数，这是因为下一个近似值需要通过在现有近似值附近进行一阶泰勒展开来确定。由微积分理论可知，任意n阶可导的函数都可在任意x_k点展开为幂函数叠加形式，故可将具有连续二阶导数的函数$f(X)$在点X_k处展开为：

$$f(X) = f(X_k) + f'(X_k)(X - X_k) + \frac{1}{2}f''(\xi)(X - X_k)^2 \qquad (2\text{-}20)$$

如果忽略上述二阶展开式的余项，则可将方程$f(X)=0$近似表示为：

$$f(X)\approx f(X_k)+f'(X_k)(X-X_k)=0$$

若$f'(X_k)\neq 0$，则可由上式得到方程$f(X)=0$的一个近似根，即$X=X_k-f(X_k)/f'(X_k)$，将其作为新的近似根，记为X_{k+1}，则可得到如下迭代式：

$$X_{k+1}=X_k-f(X_k)/f'(X_k) \qquad (2-21)$$

如果迭代初值X_0选择适当，则可通过上述迭代公式获得以方程$f(X)=0$根为极限的收敛序列$\{X_k\}$。当k值足够大时，就可获得满足精度要求的方程近似根X_k。

我们知道，对于函数优化问题，目标函数的极值点为函数驻点，即为目标函数导函数的根，故可使用上述牛顿迭代法求解目标函数导函数的根，由此获得目标函数的极值点。为此令函数$f(X)$为函数$F(X)$的导函数，则当$f(X)=0$时，$F(X)$在X点取得极值。

假设目标函数$F(X)$具有连续的三阶导数且$F''(x_k)\neq 0$，则同理可得到如下迭代式：

$$X_{k+1}=X_k-F'(x_k)/F''(x_k) \qquad (2-22)$$

适当选择初值X_0就可使上述迭代收敛到方程$F'(x)=0$的根，即目标函数$F(\mathrm{x})$的极值点，故可用这种推广的牛顿法进行模型优化。然而，机器学习的代价函数或目标函数通常比较复杂，一般会包含多个模型参数，此时通过牛顿法进行模型参数更新就相当于求解多元目标函数的极小值点。故将上述一元函数的牛顿法进一步推广到多元函数的向量情形。

设$F(X)$为三次可微的n元函数，则由多元函数泰勒展开式将其在X_k展开，得：

$$F(X)\approx F(X_k)+\nabla F(X_k)\cdot(X-X_k)+\frac{1}{2}(X-X_k)^T\cdot\nabla^2F(X_k)\cdot(X-X_k) \qquad (2-23)$$

其中$X=(x_1,x_2,\ldots,x_n)^T$，$\nabla F(X_k)$为$F(X)$在$X=X_k$处的一阶导数，即：

$$\nabla F(X_k)=\left[\frac{\partial F}{\partial x_1},\frac{\partial F}{\partial x_2},\ldots,\frac{\partial F}{\partial x_n}\right]^T_{X_k} \qquad (2-24)$$

$\nabla^2F(X_k)$为$F(X)$在$X=X_k$时的二阶导数，是一个 Hesse 矩阵，具体形式为：

$$\nabla^2F(X_k)=\begin{bmatrix}\frac{\partial^2F}{\partial x_1^2} & \frac{\partial^2F}{\partial x_1\partial x_2} & \cdots & \frac{\partial^2F}{\partial x_1\partial x_n}\\ \frac{\partial^2F}{\partial x_2\partial x_1} & \frac{\partial^2F}{\partial x_2^2} & \cdots & \frac{\partial^2F}{\partial x_2\partial x_n}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2F}{\partial x_n\partial x_1} & \frac{\partial^2F}{\partial x_n\partial x_2} & \cdots & \frac{\partial^2F}{\partial x_n^2}\end{bmatrix}_{X_k} \qquad (2-25)$$

假定$(2-23)$式右边为n元正定二次凸函数且存在唯一的最优解，对上式求一阶微分$\nabla F(X)$并令$\nabla F(X)=0$，则可将$\nabla F(X)=0$近似地表示为：

$$\nabla F(\mathrm{X})\approx\nabla F(X_k)+\nabla^2F(X_k)\cdot(X-X_k)=0$$

由上式可得$\nabla F(X)=0$的一个近似解，记为X_{k+1}，则有到如下迭代式：

$$X_{k+1}=X_k-[\nabla^2F(X_k)]^{-1}\nabla F(X_k) \qquad (2-26)$$

可将上式表示为迭代搜索通式$X_{k+1}=X_k+step_kP_k$，其中搜索步长$step_k$恒为 1，搜索方向为$P_k=-[\nabla^2F(X_k)]^{-1}\nabla F(X_k)$。由于方向$P_k$为从$X_k$到二次函数极小点的方向，故亦称之为从$X_k$发出的**牛顿方向**。由此可知，牛顿迭代法其实就是从迭代初始点开始，沿着牛顿方向且步长恒为 1 的迭代搜索算法。根据以上讨论，可得牛顿迭代法的具体计算步骤归纳如下：

（1）设定初始点X_0和终止准则，并置$X_k = 0$；

（2）求解点X_k对应的目标函数值，梯度和 Hesse 矩阵；

（3）根据$P_k = -[\nabla^2 \mathrm{F}(X_k)]^{-1}\nabla \mathrm{F}(X_k)$确定搜索方向$P_k$；

（4）依迭代公式$(2-26)$确定下一个点X_{k+1}；

（5）判断是否满足终止条件，若满足，则输出解X_{k+1}，否则$k = k + 1$，转到步骤 2。

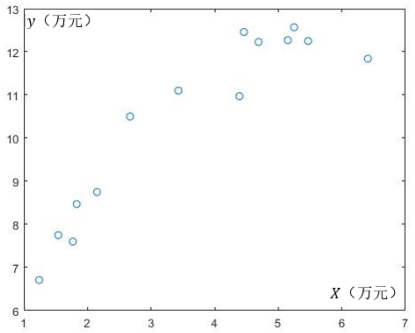


图 2-4 广告投入和净利润数据散点图

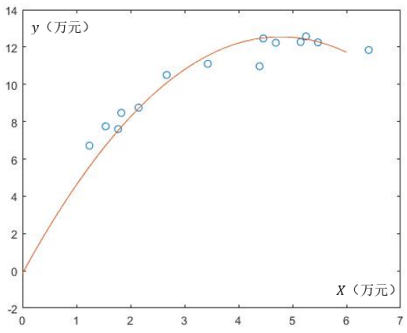


图 2-5 最终模型的函数图像

【例题 2.6】试根据表 2-11 中数据建立一个预测广告投入和净利润之间关系的机器学习模型，并使用该模型预测广告投入为 2.1 万元时所对应的净利润，要求模型优化过程采用牛顿迭代法。

表 2-11 广告投入和销售额数据表

广告投入X（万元）	4.69	6.41	5.47	3.43	4.39	2.15	1.54
净利润y（万元）	12.23	11.84	12.25	11.10	10.97	8.75	7.75
广告投入X（万元）	2.67	1.24	1.77	4.46	1.83	5.15	5.25
净利润y（万元）	10.50	6.71	7.60	12.46	8.47	12.27	12.57

【解】画出表 2-11 中数据散点图如图 2-4 所示。由图 2-4 可知，可用二次函数拟合表中数据，故设机器学习模型为$y = w_0 + w_1 X + w_2 X^2$。使用该模型构建目标函数并用误差平方和作为优化目标。为便于计算，将目标函数定义为$1/2$倍的误差平方和，即：

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{n}\left(y^i - y_p^i\right)^2 = \frac{1}{2}\sum_{i=1}^{n}\left(y^i - w_2 X^{i^2} - w_1 X^i - w_0\right)^2$$

其中y^i为第i个数据的真实值，y_p^i为对应的预测值。

代入数据求得目标函数具体表达式为：

$$E(w) = 2769.0w_1^2 + 1071.0w_1w_2 + 220.2w_1w_3 - 2580.0w_1 + 110.1w_2^2 + 50.45w_2w_3 - 567.3w_2 + 0.5w_3^2 - 145.5w_3 + 784.3$$

设置初始点$\boldsymbol{w}_0 = (w_0, w_1, w_2)^T$=$(1,1,1)^T$，求出$\nabla F(W_k), \nabla^2 F(W_k)$分别为：

$$\nabla F(\boldsymbol{w}_k) = \begin{bmatrix} 5538.9w_1 + 1071.0w_2 + 220.2w_3 - 2580.0 \\ 1071.0w_1 + 220.2w_2 + 50.45w_3 - 567.3 \\ 220.2w_1 + 50.45w_2 + \mathrm{x3} - 145.5 \end{bmatrix}$$

$$\nabla^2 F(\boldsymbol{w}_k) = \begin{bmatrix} 5538.0 & 1071.0 & 220.2 \\ 1071.0 & 220.2 & 50.45 \\ 220.2 & 50.45 & 1.0 \end{bmatrix}$$

因为目标函数为二次函数，故 Hesse 矩阵为常数。根据牛顿迭代法公式：

$$X_{k+1} = X_k - [\nabla^2 F(X_k)]^{-1} \nabla F(X_k)$$

求得$\boldsymbol{w}_1 = (-0.5559, 5.313, -0.1448)^T$，得到所求机器学习模型为：

$$y = -0.5559X^2 + 5.313X - 0.1448$$

该模型的函数图像如图 2-5 所示。将$X = 2.1$代入模型可得$y = 8.560981$，即广告投入为 2.1 万元时预测可获得的净利润为 8.560981 万元。□

牛顿法的收敛速度很快，这是其他算法难以媲美的。究其原因则是由于该算法每次迭代都会构造一个恰当的二次函数逼近目标函数，并使用从迭代点指向该二次函数极小点的方向来构造搜索方向。牛顿法的不足之处主要在于搜索方向构造困难，不仅需要计算梯度，还要计算 Hesse 矩阵及逆矩阵。为此介绍一种名为**拟牛顿法**的改进牛顿法。

拟牛顿法不仅收敛速度快，而且无需计算 Hesse 矩阵。首先给出拟牛顿法的基本原理和实现步骤，然后介绍拟牛顿法中一种有效的具体实现算法，即 DFP 算法。

令$G_k = \nabla^2 \mathrm{F}(X_k)$，$g_k = \nabla F(X_k)$, 则可将牛顿法迭代式转化为如下形式：

$$X_{k+1} = X_k + step_k G_k^{-1} g_k \qquad (2-27)$$

拟牛顿法的基本原理就是寻求一个近似矩阵来取代 Hesse 矩阵的逆矩阵G_k^{-1}。假设这个近似矩阵为$H_k = H(X_k)$, 则迭代公式可化为如下形式：

$$X_{k+1} = X_k + step_k H_k g_k \qquad (2-28)$$

显然，当近似矩阵H_k为单位阵时，则上式就是梯度法的迭代公式。为使得H_k能够更好近似G_k^{-1}，需要其满足如下条件：

1、H_k应为对称正定矩阵，以确保算法朝着目标函数下降的方向搜索；

2、H_{k+1}和H_k之间应有一定关系，例如$H_{k+1} = H_k + \Psi_k$，其中Ψ_k为修正矩阵。

3、H_k应满足拟牛顿条件。

现在导出拟牛顿条件。假设目标函数F(X)具有连续三阶导数，将F(X)在$X = X_{k+1}$处进行二阶泰勒展开并略去余项，可得：

$$F(\mathrm{X}) \approx F(X_{k+1}) + \nabla F(X_{k+1}) \cdot (X - X_{k+1}) + \frac{1}{2}(X - X_{k+1})^T \cdot \nabla^2 F(X_{k+1}) \cdot (X - X_{k+1})$$

$$\nabla F(X) = \nabla F(X_{k+1}) + \nabla^2 F(X_{k+1}) \cdot (X - X_{k+1})$$

令$X = X_k$，则有$\nabla F(X_{k+1}) = \nabla F(X_k) + \nabla^2 F(X_{k+1}) \cdot (X_{k+1} - X_k)$，亦即：

$$g_{k+1} = g_k + G_{k+1}(X_{k+1} - X_k) \qquad (2-29)$$

则当G_{k+1}正定时有$G_{k+1}^{-1} = (g_{k+1} - g_k)/(X_{k+1} - X_k)$。既然要将矩阵$H_{k+1}$近似取代$G_{k+1}^{-1}$，那么$H_{k+1}$也应该满足：

$$H_{k+1} = H_k + \Psi_k = \frac{(X_{k+1} - X_k)}{(g_{k+1} - g_k)}$$

上式即为拟牛顿条件。

令$\Delta X_k = X_{k+1} - X_k$，　$\Delta g_k = g_{k+1} - g_k$，则亦可将拟牛顿条件写成：

$$H_{k+1} = \Delta X_k / \Delta g_k \qquad (2-30)$$

拟牛顿法一定程度上保留了牛顿法计算速度较快的优势，其具体实现还取决于Ψ_k的选取，不同Ψ_k构成不同的具体算法。下面介绍一种名为DFP的常用拟牛顿法。DFP算法对拟牛顿法中修正公式的修正项进行如下优化，即设：

$$\Psi_k = \delta_k Q_k Q_k^T + \tau_k M_k M_k^T \qquad (2-31)$$

其中δ_k和τ_k都是待定常数，Q_k和M_k是n维向量。

由于上式应满足(2-23))式，故有$\Psi_k \Delta g_k = \Delta X_k - H_k \Delta g_k$，即有：

$$\delta_k Q_k Q_k^T \Delta g_k + \tau_k M_k M_k^T \Delta g_k = \Delta X_k - H_k \Delta g_k \tag{2-32}$$

令$\Delta X_k = \delta_k Q_k Q_k^T \Delta g_k$； $-H_k \Delta g_k = \tau_k M_k M_k^T \Delta g_k$；$Q_k = \Delta X_k$；$M_k = H_k \Delta g_k$，则有：

$$\delta_k = \frac{1}{\Delta {X_k}^T \Delta g_k}, \tau_k = \frac{1}{\Delta {g_k}^T H_k \Delta g_k}$$

代回(2-30)式，得到：

$$\Psi_k = \frac{\Delta X_k \Delta {X_k}^T}{\Delta {X_k}^T \Delta g_k} - \frac{H_k \Delta g_k \Delta {g_k}^T H_k}{\Delta {g_k}^T H_k \Delta g_k} \tag{2-33}$$

此时，可将修正公式转化为：

$$H_{k+1} = H_k + \frac{\Delta X_k \Delta {X_k}^T}{\Delta {X_k}^T \Delta g_k} - \frac{H_k \Delta g_k \Delta {g_k}^T H_k}{\Delta {g_k}^T H_k \Delta g_k} \tag{2-34}$$

下面结合实例介绍DFP算法具体实现过程。例如对于优化问题$\min F(\mathrm{X}) = x_1^2 + 4x_2^2$取初始点$X_0 = [1,1]^T$，则有$g_0 = [2,8]^T$。根据公式进行计算，可得：

$$X_1 = [0.73846, -0.04616]^T, g_1 = [1.47692, -0.36923]^T$$

$$\Delta X_0 = [-0.26154, -1.04616]^T, \Delta g_0 = [-0.52308, -8.36923]^T$$

由(2-34)式，可得：

$$H_1 = \begin{bmatrix} 1.00380 & -0.03149 \\ -0.03149 & 0.12697 \end{bmatrix}$$

从而算得搜索方向$P_1 = -H_1 g_1 = [-1.49416, 0.09340]^T$。通过对下列目标函数的优化计算$\mathrm{argmin}_{step_1 \geq 0} F(X_1 + step_1 P_1)$获得搜索步长，得到$step_1 = 0.49423$，由此算出$X_2 = [0,0]^T$。由于在点$X_2$处梯度为 0，故$X_2$即为最优解。

【例题 2.7】求解下面的无约束优化问题：

$$\min F(x) = 2x_1^2 + 4x_2^2 + x_3^2 - 4x_1 - 12x_2 - 2x_2 x_3 + 14$$

【解】取初始点$X_0 = [0,0,0]^T, H_0 = I$, 设置精度$e = 1 \times 10^{-12}$，当满足精度或在某点梯度为 0 时迭代终止。当$X_0 = [0,0,0]^T$时，求得$g_0 = [-4, -12, 0]^T$。确定搜索方向，得到：

$$P_0 = -H_0 g_0 = [4,12,0]^T$$

通过优化计算$\mathrm{argmin}_{step_0 \geq 0} F(X_0 + step_0 P_0)$获得搜索步长，可得到步长$step_0$和$X_1$：

$$step_0 = 0.131579; \quad X_1 = [0.526316, 1.57895, 0]^T$$

由此算得X_1点处的梯度为：

$$g_1 = [-1.89474, 0.631579, -3.15789]^T$$

从而算得$\Delta g_0, H_1$分别为：

$$\Delta g_0 = [2.10526, 12.6316, -3.15789]^T$$

$$H_1 = \begin{bmatrix} 0.98768 & -0.113393 & 0.0382166 \\ -0.113393 & 0.201224 & 0.229299 \\ 0.0382166 & 0.229299 & 0.942675 \end{bmatrix}$$

确定搜索方向：

$$P_1 = -H_1 g_1 = [2.06369, 0.382166, 2.90446]^T$$

同理计算步长以及新的点：$step_1 = 0.419146, X_2 = [1.3913, 1.73913, 1.21739]^T$。计算在$X_2$处的梯度$g_2$，梯度差$\Delta g_1$和$H_2$分别为：

$$g_2 = [1.56522, -0.521739, -1.04348]^T, \Delta g_1 = [3.45995, -1.15332, 2.11442]^T$$

$$H_2 = \begin{bmatrix} 0.335847 & -0.0572316 & -0.171695 \\ -0.0572316 & 0.204821 & 0.28113 \\ -0.171695 & 0.28113 & 1.01006 \end{bmatrix}$$

确定搜索方向

$$P_2 = -H_2 g_2 = [-0.734694, 0.489796, 1.46939]^T$$

计算步长以及新的点$step_3 = 0.532609, X_3 = [1.0, 2.0, 2.0]^T$。计算在$X_3$处的梯度$g_3 = [0, 0, 0]^T$，停止迭代。取该无约束优化问题的最优解为$X^* = X_3 = [1.0, 2.0, 2.0]^T$。□

2.3 模型优化的概率方法

机器学习领域很多模型优化方法通过概率工具实现问题求解。从直观上看，算法应该追求尽可能地稳定，引入概率方法有时破坏算法的稳定性。但机器学习领域的很多模型优化问题都需要处理巨大的求解空间或可能存在的不确定性信息，此时难以通过基本优化方法实现求解，概率方法则可以通过适当的概率工具有效地解决这些问题。因此，模型优化的概率方法是机器学习理论研究和应用开发领域非常重要的基础知识。本节主要介绍三种常用的模型优化概率方法，即随机梯度法、最大期望法和蒙特卡洛法。

2.3.1 随机梯度法

随机梯度法是对梯度下降法的一种改进。梯度下降方法在机器学习模型进行优化时，其搜索方向的每次更新都需计算所有训练样本。例如对于模型$f(X;\boldsymbol{\beta})$，其中$\boldsymbol{\beta}$为模型参数向量，即$\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_t\}$，若用训练样本集合$S = \{(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)\}$对该模型进行训练并将经验风险作为优化目标，则优化的目标函数为：

$$F(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} L(f(X_i), y_i) \qquad (2\text{-}35)$$

使用梯度下降法对上述目标函数进行优化时，第k次迭代的搜索方向计算公式为：

$$P_k = -\nabla_{\boldsymbol{\beta}} F(\boldsymbol{\beta_k}) = -\frac{1}{n}\sum_{i=1}^{n} \nabla_{\boldsymbol{\beta}} L\left(f_{\boldsymbol{\beta_k}}(X_i), y_i\right)$$

其中$\boldsymbol{\beta_k}$为第k次迭代的起始点。

由于训练样本集S包含n个样本，故在计算P_k时需分别计算这n个样本的损失函数对参数向量$\boldsymbol{\beta}$的梯度并求出它们的均值。当n较大时，计算梯度需要耗费大量时间。由于提高模型泛化性能需要尽可能多的训练样本，故不能通过减少训练样本的方式来简化计算。

事实上，在训练样本集S较大时，可用随机梯度法实现对目标函数的优化。随机梯度法的基本思想是随机选择少量训练样本计算梯度，并将该梯度作为在全部训练样本上梯度的近似代替值用于梯度下降的迭代计算。随机梯度法有很多具体的实现算法，随机梯度下降法是其中最基本也是最常用的一种。下面具体介绍随机梯度下降法。

随机梯度下降法每次迭代的方向计算只与单个样本有关。对于机器学习模型$f(X;\boldsymbol{\beta})$及

其训练样本集$S=\{(X_1,y_1),(X_2,y_2),\dots,(X_n,y_n)\}$，该方法每次随机选取$S$中单个样本代替全部样本对模型参数进行一次更新，然后换另一个未参与训练样本进行下次更新，当S中全部样本都参与更新计算之后，随机调整S中所有样本排列次序后重复以上过程。

具体地说，假设随机梯度下降法在进行第k次迭代时随机选择的样本为(X_i,y_i)，则此次参数更新的搜索方向可通过该样本的损失函数计算得到。令$\boldsymbol{\beta}_k$为模型的当前参数向量，则用函数$F'(\boldsymbol{\beta})=L\left(f_{\boldsymbol{\beta}_k}(X_i),y_i\right)$实现此次迭代搜索方向的更新。需要注意的是，函数$F'(\boldsymbol{\beta})$为随机选择到的样本$(X_i,y_i)$的损失函数，仅用于计算参数搜索的更新方向，模型优化的目标函数则保持不变。方向更新的具体计算公式为：

$$P'_k=-\nabla_{\boldsymbol{\beta}}F'(\boldsymbol{\beta})=-\nabla_{\boldsymbol{\beta}}L\left(f_{\boldsymbol{\beta}_k}(X_i),y_i\right)$$

显然，使用上述公式仅需计算一次梯度便可实现方向更新。将P'_k代替P_k，并进行第k次参数更新的结果为：

$$\boldsymbol{\beta}_{k+1}=\boldsymbol{\beta}_k-step_k\nabla_{\boldsymbol{\beta}}L\left(f_{\boldsymbol{\beta}_k}(X_i),y_i\right) \qquad (2-36)$$

由以上分析可知：梯度下降法在n元训练样本集S上对模型进行优化时，每次参数更新需要执行n次的梯度计算。使用随机梯度下降法则只需执行一次梯度计算，可以有效减小计算量，使得模型优化过程在大样本场合变得可行。随机梯度下降法的具体过程如下：

（1） 初始化$t=0$，$k=0$；

（2） 对S中的全部训练样本进行随机排序，得到新的样本集合S^t；

（3） 若S^t中样本均参与过训练过程，则令$t=t+1$，并重复步骤 2，否则随机选择S^t中未参与训练的单个样本，并根据该样本数据计算参数更新方向P_k；

（4） 根据公式（2-30）更新参数向量得第k次参数更新得结果$\boldsymbol{\beta}_{k+1}$；

（5） 判断迭代是否满足算法终止条件，若满足则终止算法，否则令$k=k+1$, 重复步骤 3、4；

事实上，目前的计算机多采用多核架构，具有一定的并行计算能力，若每次只使用单个样本确定参数搜索方向的更新，则会造成计算能力的浪费。因此，小批量随机梯度下降法也是一种常用的随机梯度法。

小批量随机梯度下降法的基本步骤与随机梯度下降法类似，基本思想是首先从S中随机选择一小批训练样本代替全部样本实现对模型参数搜索方向的更新，然后在下次迭代搜索中再换另一小批不同样本进行搜索方向的更新计算，当训练样本集S中的全部样本都参与过一次参数的更新之后，将S中的全部样本进行随机排序后重新划分成不同小批次去更新搜索方向，直至目标函数的取值接近最小值。显然，随机梯度下降法可以看作是小批量随机梯度下降法在每个小批量样本数均为 1 的一种特殊情况。

现结合实例说明小批量随机梯度下降法的具体过程。假设训练集S共有$m=100000$个样本，将其中全部样本随机排序后得到集合S'，即有：

$$S'=\{(X'_1,y'_1),(X'_2,y'_2),\dots,(X'_m,y'_m)\}$$

首先，将S'划分为若干小批量训练样本集，例如将S'等分为 1000 个规模为 100 的小批量训练样本集$\mathrm{S}_1,\mathrm{S}_2,\dots,\mathrm{S}_{1000}$, 即有：

第 1 个小批量训练集$\mathrm{S}_1=\{(X'_1,y'_1),\dots,(X'_{100},y'_{100})\}$

第 2 个小批量训练集$\mathrm{S}_2=\{(X'_{101},y'_{101}),\dots,(X'_{200},y'_{200})\}$

……

第 1000 个小批量训练集$S_{1000}=\{(X'_{99901},y'_{99901}),\cdots,(X'_{100000},y'_{100000})\}$

然后，在每次迭代过程随机选择一个当前未参与训练的小批次进行搜索方向的更新计算。假设当前模型参数向量为$\boldsymbol{\beta}_k$并选择第i个批次对进行方向更新，则有：

$$F'(\boldsymbol{\beta})=\frac{1}{100}\sum_{X'_j\in S_i}L\left(f_{\boldsymbol{\beta}_k}(X'_j),y'_j\right) \qquad (2-37)$$

据此计算更新方向为：

$$P'_k=-\nabla_{\boldsymbol{\beta}}F'(\boldsymbol{\beta})=-\frac{1}{100}\sum_{X'_j\in S_i}\nabla_{\boldsymbol{\beta}}L\left(f_{\boldsymbol{\beta}_k}(X'_j),y'_j\right)$$

依据上述更新方向对当前参数向量$\boldsymbol{\beta}_k$进行更新：

$$\boldsymbol{\beta}_{k+1}=\boldsymbol{\beta}_k-step_k\frac{1}{100}\sum_{X'_j\in S_i}\nabla_{\boldsymbol{\beta}}L\left(f_{\boldsymbol{\beta}_k}(X'_j),y'_j\right) \qquad (2-38)$$

当所有划分的批次均参与了训练过程而未达到算法终止条件时，对S'进行随机排序后重新划分成若干小批次样本集合，并重复上述搜索方向和参数更新过程，直至达到算法终止条件时结束算法。

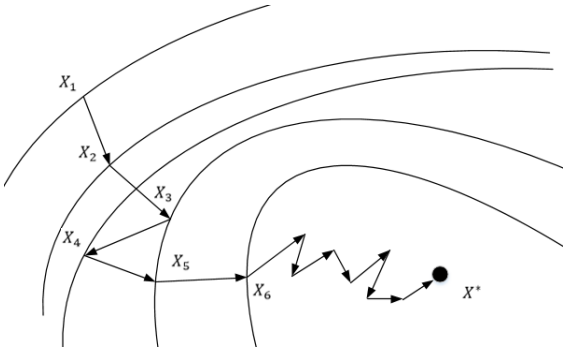


图 2-6 随机梯度法参数更新示意图

随机梯度法每次仅选择单个或小批量样本确定搜索方向和实现对模型参数的更新，可有效减少花费在梯度计算上的时间。该方法的计算结果存在一定的随机性，即不能保证每次更新的方向都是目标函数减小的方向，参数更新偶尔会使得目标函数取值增大。然而，取自相同总体的样本服从相同的概率分布，并且所有训练样本均参与迭代计算，故在参数更新的整体过程中，随机梯度下降法可以保证目标函数的取值不断下降。随机梯度法的参数更新过程如图 2-6 所示。

另一方面，随机梯度方法的参数更新方向存在着一定的随机性，为避免陷入局部最优提供了可能。正常情况下梯度下降算法很难跳出局部最优，但是随机梯度法打破了参数更新的稳定性，这会使得迭代过程可能出现跳出当前的局部最小值范围。

目前，很多机器学习任务的训练样本集规模都很大，由以上分析可知随机梯度方法能够在保证优化效果的前提下有效减少在大规模训练样本集上对模型进行优化的计算量。因此，随机梯度法是目前最常用的模型优化方法之一。

2.3.2 最大期望法

有些机器学习模型含有隐含变量，通常难以对这些具有隐含变量的模型直接进行参数求解或估计，此时可通过迭代求解隐含变量（或其函数）数学期望最大值的方法实现对带隐含变量模型的参数优化求解。这类优化方法通常称之为最大期望法或EM算法，主要包括两个计算步骤，即计算数学期望的E步骤和函数最大值优化计算的M的步骤。可用EM算法解决带隐含变量模型参数的最大似然估计或者最大后验估计问题。这里主要讨论如何通过EM算法求解参数的最大似然估计，最大后验估计的EM求解方法与此类似。

假设某学校男生和女生身高分别服从参数不同的两个高斯分布，现从该学校所有学生当中随机抽取 100 名学生，分别测量他们的身高数据但不记录性别信息。根据这些数据对男生身高和女生身高所服从高斯分布进行参数估计时，则存在一个隐含的性别数据。由于不知道性别信息，无法直接得知性别数据的分布，故无法直接求得已知样本数据所满足的似然函数。然而，在给定参数情况下，可以结合已知样本计算出某个同学性别的概率分布情况，并且可以知道在给定参数情况下性别与身高的联合概率分布。

更一般的，在一个包含隐含数据Z的模型参数估计问题中，X为可直接观测数据，即已知样本，$\boldsymbol{\beta}$为模型参数向量。由于隐含数据Z的存在，通常无法直接得知已知样本X取值下的似然函数$L(\boldsymbol{\beta}|X)=p(X|\boldsymbol{\beta})$，但可知道参数给定情况下$X$和$Z$的联合概率分布$p(X,Z|\boldsymbol{\beta})$，以及在参数向量和可观测数据给定情况下$Z$取值状态的条件概率分布$p(Z|X,\boldsymbol{\beta})$。

现根据以上信息对模型参数向量$\boldsymbol{\beta}$进行最大似然估计。最大似然估计通过最大化似然函数实现，然而已知样本X取值状态的似然函数为$L(\boldsymbol{\beta}|\mathrm{X})=p(X|\boldsymbol{\beta})$却由于隐含数据Z的存在而难以直接求得，故难以直接使用似然函数对模型参数向量$\boldsymbol{\beta}$进行最大似然估计。由于已知参数$\boldsymbol{\beta}$给定条件下X和Z的联合概率$p(X,Z|\boldsymbol{\beta})$，故将上述似然函数转化为：

$$L(\boldsymbol{\beta}|\mathrm{X})=\int p(X,Z|\boldsymbol{\beta})dZ \qquad (2-39)$$

考虑其对数似然，有：

$$\ln L(\boldsymbol{\beta}|X)=\ln\int p(X,Z|\boldsymbol{\beta})dZ \qquad (2-40)$$

根据最大似然估计思想，只需求得对数似然$\ln L(\boldsymbol{\beta}|X)$最大值即可得到模型参数$\boldsymbol{\beta}$的最大似然估计值。但是由于上式存在隐含数据$Z$和积分的对数，直接求解其最大值较为困难，故用迭代逼近方法实现对数似然最大值的估计。为此，将对数似然做如下变形：

$$\ln L(\boldsymbol{\beta}|X)=\ln\int\frac{p(X,Z|\boldsymbol{\beta})}{p(Z)}p(Z)dZ \qquad (2-41)$$

其中$p(Z)$为隐含数据Z的某一分布。

由于对数函数为凸函数，故成立如下不等式：

$$\ln L(\boldsymbol{\beta}|X)=\ln\int\frac{p(X,Z|\boldsymbol{\beta})}{p(Z)}p(Z)dZ\geq\int\ln\left[\frac{p(X,Z|\boldsymbol{\beta})}{p(Z)}p(Z)\right]dZ$$

由此可得对数似然的下界函数：

$$B(\boldsymbol{\beta})=\int\ln\left[\frac{p(X,Z|\boldsymbol{\beta})}{p(Z)}p(Z)\right]dZ \qquad (2-42)$$

取$p(Z)=p(Z|X,\boldsymbol{\beta}_t)$, 则可得：

$$B(\boldsymbol{\beta},\boldsymbol{\beta}_t)=\int \ln\left[\frac{p(X,Z|\boldsymbol{\beta})}{p(Z)}p(Z)\right]dZ=\int \ln\left[\frac{p(X,Z|\boldsymbol{\beta})}{p(Z|X,\boldsymbol{\beta}_t)}p(Z|X,\boldsymbol{\beta}_t)\right]dZ$$

$$=\int \ln p(X,Z|\boldsymbol{\beta})\,p(Z|X,\boldsymbol{\beta}_t)dZ-\int \ln p(Z|X,\boldsymbol{\beta}_t)\,p(Z|X,\boldsymbol{\beta}_t)dZ$$

略去下界函数$B(\boldsymbol{\beta},\boldsymbol{\beta}_t)$中与待求参数向量 $\boldsymbol{\beta}$ 无关的项，得到如下Q函数：

$$Q(\boldsymbol{\beta},\boldsymbol{\beta}_t)=\int \ln L(\boldsymbol{\beta}|X,Z)p(Z|X,\boldsymbol{\beta}_t)dZ \qquad (2\text{-}43)$$

上式表示对隐含变量Z的函数$L(\boldsymbol{\beta}|X,Z)$在概率分布$p(Z|X,\boldsymbol{\beta}_t)$下的数学期望。由于$B(\boldsymbol{\beta},\boldsymbol{\beta}_t)\le \ln L(\boldsymbol{\beta}|X)$，故可通过迭代选取不同下界函数$B(\boldsymbol{\beta},\boldsymbol{\beta}_t)$最大值的方式逐步逼近对数似然$\ln L(\boldsymbol{\beta}|X)$的最大值，迭代逼近的具体过程如图 2-7 所示。

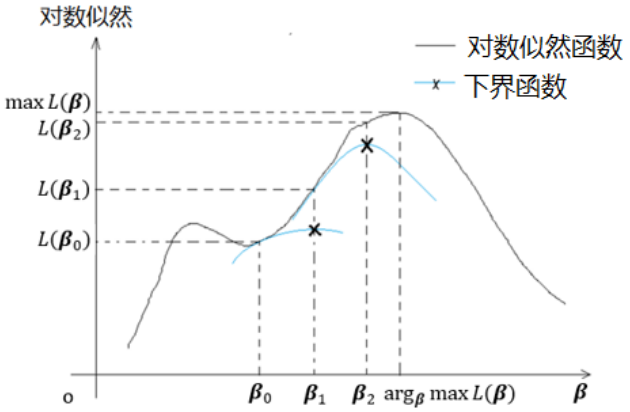


图 2-7 EM算法逼近似然函数最大值

由于$Q(\boldsymbol{\beta},\boldsymbol{\beta}_t)$通过省略$B(\boldsymbol{\beta},\boldsymbol{\beta}_t)$中与待求参向量$\boldsymbol{\beta}$无关项得到，用$Q(\boldsymbol{\beta},\boldsymbol{\beta}_t)$迭代求解对数似然最大值与用$B(\boldsymbol{\beta},\boldsymbol{\beta}_t)$求解对数似然最大值具有相同的效果，故EM算法通常直接使用Q函数进行优化计算。由以上分析可得EM算法的基本步骤如下：

(1) 设置初始参数$\boldsymbol{\beta}_0$和迭代停止条件；

(2) E步（期望步）：根据可直接观测数据X和当前参数向量取值$\boldsymbol{\beta}_t$计算 $Q(\boldsymbol{\beta},\boldsymbol{\beta}_t)$

$$\mathrm{Q}(\boldsymbol{\beta},\boldsymbol{\beta}_t)=\int \ln L(\boldsymbol{\beta}|\mathrm{Y})p(Z|X,\boldsymbol{\beta}_t)dZ$$

(3) M步（最大化步）：最大化$Q(\boldsymbol{\beta},\boldsymbol{\beta}_t)$并根据$Q(\boldsymbol{\beta},\boldsymbol{\beta}_t)$最大值更新参数$\boldsymbol{\beta}_t$的取值：

$$\boldsymbol{\beta}_{t+1}=\arg_{\boldsymbol{\beta}}\max Q(\boldsymbol{\beta},\boldsymbol{\beta}_t)$$

(4) 判断是否满足迭代停止条件，若满足则停止迭代，否则令$t=t+1$并返回步骤(2)。

EM算法可随机选择初始参数$\boldsymbol{\beta}_0$，但应注意EM算法对初始参数$\boldsymbol{\beta}_0$具有一定的敏感性。如果初值$\boldsymbol{\beta}_0$选取不当，则可能会使迭代结果陷入局部最优。

通常将迭代停止条件设为相邻两次参数值变化不大或前后两次参数更新使得$Q(\boldsymbol{\beta},\boldsymbol{\beta}_t)$值变化很小时停止更新，即有：

$$|\boldsymbol{\beta}_{t+1}-\boldsymbol{\beta}_t|<\varepsilon \text{或} |Q(\boldsymbol{\beta}_{t+1},\boldsymbol{\beta}_t)-Q(\boldsymbol{\beta}_t,\boldsymbol{\beta}_t)|<\varepsilon$$

【例题 2.8】假设X_1，X_2取自指数分布$Y(\theta)=\theta e^{-x\theta}$，且$X_1$，$X_2$不相关。若$X_1=10$而$X_2$缺失，试用EM算法实现参数$\theta$的最大似然估计。

【解】由于X_1，X_2为取自指数分布$\mathrm{Y}(\theta)=\theta \mathrm{e}^{-x\theta}$的离散值，则有：

$$\ln L(\theta|X_1,X_2)=\ln p(X_1,X_2|\theta)=\ln(\theta^2 e^{-X_1\theta}e^{-X_2\theta})$$

$$= 2\ln\theta - X_1\theta - X_2\theta$$

由于X_1，X_2同分布，则有$p(X_2|X_1,\theta_t) = \theta_t \mathrm{e}^{-X_2\theta_t}$，从而有：

$$\mathrm{Q}(\theta,\theta_t) = \int \ln p(X_1,X_2|\theta)\, p(X_2|X_1,\theta_t) dX_2$$

$$= \int (2\ln\theta - X_1\theta - X_2\theta)\theta_t \mathrm{e}^{-X_2\theta_t}\, dX_2$$

$$= 2\ln\theta - X_1\theta - \theta \int X_2\theta_t \mathrm{e}^{-X_2\theta_t}\, dX_2$$

$$= 2\ln\theta - X_1 - \theta/\theta_t$$

为求使得$Q(\theta,\theta_t)$最大化的参数θ，作为第$t+1$次的估计值θ_{t+1}，即得迭代公式：

$$\theta_{t+1} = \arg_\theta \max Q(\theta,\theta_t) = 2\theta_t/(10\theta_t + 1)$$

当迭代收敛时，有$\theta_* = 2\theta_*/(10\theta_* + 1)$，解得$\theta_* = 0.1$，即迭代算法求得参数估计值收敛于 0.1，即参数$\theta$的最大似然估计值为$\theta_* = 0.1$。□

2.3.3 蒙特卡洛法

蒙特卡洛法是一类以概率统计理论为基础的随机型数值算法。一般来说，当一个随机算法满足采样次数越多，其输出结果越近似最优解这一特性时，便可称之为蒙特卡洛法。该方法通常将待求解问题与某一概率模型联系起来，利用从大量样本中获得的信息完成对所求参数的概率估计，由此实现对实际问题的求解。

例如，可用蒙特卡洛法近似计算一个不规则湖面的面积。假设围住湖面的长方形面积为M，湖面面积S为未知数。由于湖面形状不规则无法直接求出S，为此向包含湖面的长方形区域内随机撒布n个点，假设其中有k个点落在湖面当中，则可得到湖面大小S的估计值：

$$\hat{S} = Mk/n \qquad (2-44)$$

显然，当撒布点数n越大，估计值$\hat{S}$就越接近于湖面真实面积S。这是因为根据大数定律和中心极限定理，重复进行大量实验时，事件A出现的频率接近事件A发生的概率。

蒙特卡洛法最早用于近似求解难以精确计算的定积分，对于某函数$\varphi(x)$的积分：

$$I = \int_a^b \varphi(x)\, dx$$

若能找到定义在(a,b)上的一个函数$f(x)$和概率密度函数$p(x)$，满足$\varphi(x) = f(x)p(x)$，则可将上述积分I转化为：

$$I = \int_a^b f(x)p(x)\, dx = E[f(x_i)]$$

如此一来，就将原积分I的求解问题转化为了求解函数$f(x)$在$p(x)$分布上的数学期望问题。假设在分布$p(x)$上做随机采样得到的样本集合为$\{x_1, x_2, \ldots, x_n\}$，则可用这些样本来对$I$进行估计，即有：

$$I_p = \frac{1}{n}\sum_{i=1}^{n} f(x_i) \qquad (2-45)$$

由于

$$E(I_p)=\frac{1}{n}\sum_{i=1}^{n}E[f(x_i)]=\frac{1}{n}\sum_{i=1}^{n}I=I$$

故I_p是I的无偏估计，即对于任意分布$p(x)$，积分I的蒙特卡洛法估计值是无偏的。

有时在分布$p(x)$上进行采样会出现一些问题，例如采样偏差较大等，甚至在某些时候根本无法对其进行采样。此时，可通过变换积分的分解形式找到另外一个易于采样的分布$q(x)$，将积分I转化为如下形式：

$$I=\int_a^b g(x)\,dx=\int_a^b f(x)p(x)\,dx=\int_a^b f(x)\frac{p(x)}{q(x)}q(x)\,dx$$

其中$q(x)$为某一分布，可将$f(x)[p(x)/q(x)]$看作是某一函数。

令$g(\mathrm{x})=f(x)p(x)/q(x)$，则有：

$$I=\int_a^b f(x)p(x)\,dx=\int_a^b g(x)q(x)\,dx$$

此时可将积分I看作是函数$g(x)$在分布$q(x)$上的期望。假设在分布$q(x)$上进行随机采样获得的样本集合为$\{x_1',x_2',\ldots,x_n'\}$，则可用该样本集合对$I$进行估计，即有：

$$I_q=\frac{1}{n}\sum_{i=1}^{n}g(x_i')=\frac{1}{n}\sum_{i=1}^{n}\frac{p(x_i')f(x_i')}{q(x_i')}$$

在新分布$q(x)$上进行采样的过程被称为**重要性采样**，$p(x)/q(x)$被称为**重要性权重**。由于积分I任意分解形式通过蒙特卡洛法得到的估计值都是无偏的，故应适当选择新分布$q(x)$，使得估计值I_q的方差尽可能地达到最小。对于I_q的方差$\mathrm{var}(I_q)$：

$$\mathrm{var}(I_q)=\mathrm{E}_q\left(f^2(x)p^2(x)/q^2(x)\right)-I^2 \qquad (2-46)$$

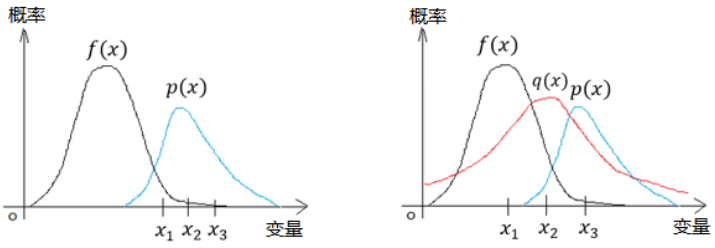
由于其后项与分布$q(x)$无关，故只需最小化期望$\mathrm{E}_q\left(f^2(x)p^2(x)/q^2(x)\right)$。

根据琴生不等式，有：

$$\mathrm{E}_q\left(f^2(x)p^2(x)/q^2(x)\right)\geq \mathrm{E}_q{}^2(f(x)p(x)/q(x))=(\int|f(x)|p(x)\,dx)^2$$

故取$q(x)$为如下分布时可使得方差$\mathrm{var}(I_q)$达到最小：

$$q^*(x)=\frac{|f(x)|p(x)}{\int|f(x)|p(x)\,dx} \qquad (2-47)$$



(a) 在分布$\boldsymbol{p}(\mathbf{x})$上进行采样　　(b) 在分布$\boldsymbol{q}(\mathbf{x})$上进行采样

图 2-8 不同采样方式对比示意图

重要性采样可以改进原采样，如图 2-8（a）所示，若在分布$p(x)$上进行采样，则很可能仅仅采样到函数$f(x)$的极端值；如图 2-8（b）所示，若在分布$q(x)$上进行重要性采样，则函数$f(x)$的取值较为丰富。

在实际应用当中，重要性采样并非总能奏效。在很多情况下，重要性采样$q(x)$难以取得理论上最佳分布$q^*(x)$，而只能取到一个方差较大的可行分布，难以满足任务需求。此时用**马尔可夫链蒙特卡洛法（MCMC）**近似计算待求参数概率分布方式解决。

假设离散型随机变量X的取值范围是$\{X_1, X_2, \dots, X_n\}$，则称该集合为X的状态空间。**马尔可夫链**是一个关于离散型随机变量取值状态的数列，从X的随机初始状态X_i开始，马尔可夫链依据一个只与前一时序状态相关的状态转移分布$P(X^{t+1}|X^t)$确定下一时序的状态，其中X^t和X^{t+1}分别表示随机变量X在第t时序和第$t+1$时序的状态。

MCMC法同时运行多条马尔可夫链，这些马尔可夫链在同一时序内具有相同的状态空间，即服从同一个状态概率分布。随着马尔可夫链的不断更新，其状态概率分布最终会收敛于某一分布。具体来说，MCMC法从一个初始概率分布p^0中随机选择多条马尔可夫链的初始状态，p^0的具体形式为：

$$p^0(X = X_i) = p_i^0 \tag{2-48}$$

其中p_i^0表示在时骤为 0 的初始状态分布下$X = X_i$的概率。

由于这些马尔可夫链的状态空间是一个n维空间，故可将概率分布p^0表示为一个n维向量$\boldsymbol{p}^0 = (p_1^0, p_2^0, \dots, p_n^0)^T$。此时，马尔可夫链下一时序的分布可表示为：

$$p^{t+1} = \sum_X p^t \, P(X^{t+1}|X^t) \tag{2-49}$$

上式说明马尔可夫链在第$t+1$时序的状态分布只与前一时序的状态分布p^t和状态转移分布$P(X^{t+1}|X^t)$相关。

接着考虑所有同时运行的马尔可夫链状态概率分布情况。若用矩阵$\boldsymbol{A}$表示$P(X^{t+1}|X^t)$，矩阵$\boldsymbol{A}$中第i行第j列元素a_{ij}表示从状态X_j转移到状态X_i的概率$P(X_i|X_j)$，则这些马尔可夫链在同一时序上状态概率分布的变化可表示为：

$$\boldsymbol{p}^{t+1} = \mathbf{A}\boldsymbol{p}^t \tag{2-50}$$

状态概率分布会随着时序t不断变大而最终收敛，即有$\boldsymbol{p}^{t+1} = \boldsymbol{p}^t$。此时可用所求状态概率分布作为真实状态概率分布的一种近似分布。

事实上，由于贝叶斯学派认为模型的未知参数服从于某一概率分布，并将参数的概率分布看作一种状态概率分布，故可通过MCMC法求解其近似分布的方式实现模型优化。

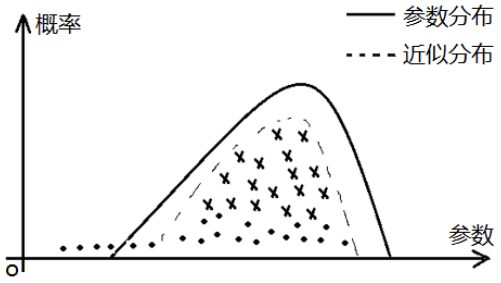


图 2-9 MCMC法的基本思想

MCMC法的基本思想如图 2-8 所示。对于参数的概率分布，$MCMC$法从随机初始点开始进行采样，随着采样的不断进行，其样本点的概率分布会逐步收敛。图中圆点表示概率分布还未收敛时采样到的点，叉点表示概率分布收敛后采样到的点。由于最终收敛的概率分布是参数概率分布的一种近似，故在其中进行大量采样便可求出该近似概率分布，即求得参数概

率分布的一种近似分布。

显然，MCMC法的关键在于如何确定状态转移概率分布$P(X^{t+1}|X^t)$，如果$P(X^{t+1}|X^t)$的选择合适，则状态概率分布会收敛到待求参数的概率分布。从收敛性角度考虑，$MCMC$法的状态转移概率分布$P(X^{t+1}|X^t)$应保证多条马尔可夫链同时满足状态分布收敛的条件。

事实上，当状态概率分布$p(X)$与状态转移概率分布$P(X^{t+1}|X^t)$满足细致平稳条件时，状态转移分布$P(X^{t+1}|X^t)$就可使得状态概率分布收敛，此时状态分布$p(X)$称为状态转移分布$P(X^{t+1}|X^t)$的**平稳分布**。

所谓**细致平稳条件**，是指在状态分布p(X)中，利用状态转移分布对其状态进行更新时，由X_i转移到X_j的概率应与由X_j转移到X_i的概率相同，即有$p(X_i)P(X_j|X_i) = p(X_j)P(X_i|X_j)$。

在实际应用中找到一个状态转移分布使其满足细致平稳条件有时是一件比较困难的事情，通常需要使用某些采样方法构造一个满足条件的状态转移分布。其中最简单的采样方法为MCMC采样。具体地说，对于状态分布p(X)，可随机选择一个状态转移分布$Q(X^{t+1}|X^t)$，它们通常不满足细致平稳条件，即有：

$$p(X_i)Q(X_j|X_i) \neq p(X_j)Q(X_i|X_j)$$

但可选择一对转移算子α(X_i, X_j)和α(X_j, X_i)满足下列等式：

$$p(X_i)Q(X_j|X_i)\alpha(X_i, X_j) = p(X_j)Q(X_i|X_j)\alpha(X_j, X_i) \qquad (2-51)$$

可用转移算子$\alpha(X_i, X_j)$和$\alpha(X_j, X_i)$帮助确定是否接受采样结果。要想使得上述等式成立，最简单的方法是令：

$$\alpha(X_i, X_j) = \mathrm{p}(X_j)\mathrm{Q}(X_i|X_j),\ \alpha(X_j, X_i) = p(X_i)Q(X_j|X_i)$$

并取状态转移分布为$P(X_i|X_j) = Q(X_j|X_i)\alpha(X_i, X_j)$，则有状态分布$p$(X)即为$P(X_i|X_j)$的平稳分布，此时就可通过MCMC方法将获得一个收敛的状态概率分布。

MCMC采样的基本步骤为：

（1） 随机选定一个状态转移分布$Q(X_j|X_i)$并记状态分布为p(X)，设定收敛前状态转移次数上限τ和采样个数n；

（2） 从任意分布中采样获得初始状态X_0；

（3） 依据条件状态分布$Q(X_j|X_i)$进行采样，即采样值为$X^{t+1} = Q(X|X^t)$；

（4） 从均匀分布U[0,1]中采样并记为u_t，若$u_t < \alpha(X^t, X^{t+1}) = p(X^t)Q(X^{t+1}|X^t)$，则接受$X^{t+1}$，并令$t = t + 1$，否则拒绝此次转移并保持$t$值不变；

（5） 当$t > \tau + n$时终止算法。此时共取到$\tau + n$个样本，但前τ个样本在分布未收敛时获得，不能作为MCMC采样结果，故将第$\tau + 1$到$\tau + $n个样本作为采样结果。

除了MCMC采样之外，还有M-H采样、Gibbs采样等获取样本方法。其中M-H采样是对MCMC采样的改进，Gibbs采样是对M-H采样的改进，这里不再赘述。

2.4 模型正则化策略

如前所述，在机器学习模型的训练过程中，经常会出现训练误差较小但泛化误差较大的过拟合现象。这是因为如果模型参数个数较多或取值范围较大而训练样本数量较少，模型中某些参数就会失去应有的约束，使得训练获得的模型较为复杂，降低了模型的泛化能力。由

此可知，产生过拟合现象的根本原因在于模型容量与训练样本数量不匹配。此时可以通过适当方法对模型的自由度或容量进行一定程度的控制，以尽量避免模型的过拟合现象。这些方法通常统称之为模型的**正则化策略**。具体可分别从模型修正和样本扩充这两个角度设计模型的正则化策略。模型正则化的目标是提升模型性能，故可将其看成是实现模型优化的一种特殊方式。本节主要介绍范数惩罚、样本增强和对抗训练这三种常用的模型正则化策略。其中范数惩罚策略基于对模型的修正，样本增强和对抗训练策略基于对样本的扩充。

2.4.1 范数惩罚

范数惩罚是机器学习领域最常用一种的模型正则化方法。该方法的基本思想是通过在目标函数表达式中添加适当惩罚项的方式降低模型容量，使得模型容量与训练样本数量相匹配，从而达到提升模型泛化性能的目的。

具体地说，假设使用基数为n的训练样本集$\mathrm{S}=\{(X_1,y_1),(X_2,y_2),\ldots,(X_n,y_n)\}$进行模型训练且目标函数为：

$$F(\boldsymbol{\beta})=\frac{1}{n}\sum_{X_i\in S}L(f(X_i),y_i) \qquad (2-52)$$

其中$\boldsymbol{\beta}$为模型参数向量，即$\boldsymbol{\beta}=(\beta_1,\beta_2,\ldots,\beta_k)^T$。

范数惩罚则在上述目标函数中添加惩罚项$\alpha\lambda(\boldsymbol{\beta})$，建立新的目标函数$F'(\boldsymbol{\beta})$，即有：

$$F'(\boldsymbol{\beta})=\frac{1}{n}\sum_{X_i\in S}L(f(X_i),y_i)+\alpha\lambda(\boldsymbol{\beta}) \qquad (2-53)$$

惩罚项$\alpha\lambda(\boldsymbol{\beta})$中$\alpha$为大于 0 的超参数，$\alpha$越大，则对参数的惩罚越严重。$\lambda(\boldsymbol{\beta})$表示对参数向量$\boldsymbol{\beta}$的惩罚形式，根据其形式不同可将范数惩罚分为多种具体形式。

常用的范数惩罚方式主要有L^1范数惩罚、L^2范数惩罚等。L^1范数惩罚是对目标函数$F(\boldsymbol{\beta})$添加L^1范数形式的正则化项：

$$\tau=\alpha\|\boldsymbol{\beta}\|_1=\alpha\sum_{i=1}^{n}|\beta_i|$$

其中α为正则化权重，模型越复杂时α的值应该设置得越大。正则化后目标函数为：

$$F'(\boldsymbol{\beta})=\frac{1}{n}\sum_{X_i\in S}L(f(X_i),y_i)+\alpha\|\boldsymbol{\beta}\|_1 \qquad (2-54)$$

可用梯度下降算法最小化目标函数$F'(\boldsymbol{\beta})$。对$F'(\boldsymbol{\beta})$计算梯度：

$$\nabla F'(\boldsymbol{\beta})=\frac{1}{n}\sum_{X_i\in S}\nabla_{\boldsymbol{\beta}}L(f(X_i),y_i)+\alpha\mathrm{sign}(\boldsymbol{\beta}) \qquad (2-55)$$

其中$\mathrm{sign}(x)$为符号函数，即对于任意参数β_i，有：

$$\mathrm{sign}(\beta_i)=\begin{cases}1,\beta_i>0\\0,\beta_i=0\\-1,\beta_i<0\end{cases}$$

使用梯度下降算法进行参数更新，即有：

$$\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^{t} - \varepsilon\alpha\text{sign}(\boldsymbol{\beta}^{t}) - \varepsilon\nabla F(\boldsymbol{\beta}^{t}) \qquad (2\text{-}56)$$

上式表明，带L^1范数惩罚项的梯度下降法在原始梯度下降算法的基础之上，对参数的取值做一定程度的控制，使其尽量向 0 靠近，即若$\beta_i > 0$,则减去$|\varepsilon\alpha\text{sign}(\beta_i)|$，若$\beta_i < 0$,则加上$|\varepsilon\alpha\text{sign}(\beta_i)|$。因此，$L^1$范数惩罚的基本目的是尽量产生稀疏的参数向量，即使得尽可能多的参数值为 0。

模型容量过大通常是由于参数过多， L^1范数惩罚将很多模型参数置为 0。这意味着降低了模型对数据的拟合能力，控制了模型容量，由此可缓解过拟合现象。

此外，由于L^1范数惩罚项τ的表达式中包含绝对值，故目标函数$F'(\boldsymbol{\beta})$并非处处可导。这相当于对原目标函数$F(\boldsymbol{\beta})$的性质做出约束。图 2-10 表示二维参数向量在L^1范数惩罚下的约束效果，其中等值线为目标函数图像，而方型线为惩罚项$\tau = |\beta_1| + |\beta_2|$的图像。

由于惩罚项τ在某些点不可导，故其图像会出现尖点。显然，惩罚项τ不可导点的坐标中至少有一项为 0，在高维情况下可能有多项为 0。因此， L^1范数惩罚通常可使参数向量稀疏。从特征选择的角度看，参数向量稀疏相当于对模型的输入特征进行了一定的优化选择，即L^1范数惩罚保留了与模型输出相关的特征而排除了无关特征。

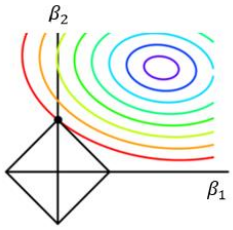


图 2-10 $\boldsymbol{L^1}$范数惩罚示意图

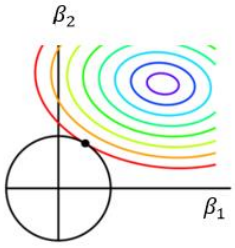


图 2-11 $\boldsymbol{L^2}$范数惩罚示意图

与L^1范数惩罚类似，L^2范数惩罚对目标函数$F(\boldsymbol{\beta})$添加L^2范数的正则化项：

$$\varphi = \frac{\alpha}{2}\|\boldsymbol{\beta}\|_2^2 = \frac{\alpha}{2}(\sum_{i=1}^{n}\beta_i^2)$$

得到新的目标函数$F'(\boldsymbol{\beta})$，即有：

$$F'(\boldsymbol{\beta}) = \frac{1}{n}\sum_{X_i \in S} L(f(X_i), y_i) + \frac{\alpha}{2}\|\boldsymbol{\beta}\|_2^2 \qquad (2\text{-}57)$$

可用梯度下降算法对目标函数$F'(\boldsymbol{\beta})$的最小值优化。计算$F'(\boldsymbol{\beta})$的梯度：

$$\nabla F'(\boldsymbol{\beta}) = \nabla \text{F}(\boldsymbol{\beta}) + \alpha\boldsymbol{\beta}$$

得到新的迭代公式为：

$$\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^{t} - \varepsilon\left(\alpha\boldsymbol{\beta}^{t} + \nabla F(\boldsymbol{\beta}^{t})\right)$$

即有：

$$\boldsymbol{\beta}^{t+1}(1 - \varepsilon\alpha)\boldsymbol{\beta}^{t} - \varepsilon\nabla F(\boldsymbol{\beta}^{t}) \qquad (2\text{-}58)$$

其中ε为梯度下降算法的步长。

由于ε和α均大于 0，故使用L^2范数惩罚所得参数是在原参数缩小$\varepsilon\alpha$倍的基础上计算获得的。这种方法通常称之为**权重衰减**。由此可见，L^2范数惩罚并没有刻意促使某些参数为 0，

而是使得模型的所有参数值都变小。

一般而言，过拟合模型都较为复杂且参数值。例如，对于图 2-12 所示的回归模型，由于该模型试图充分拟合每个样本点，这使得其导数值较大。当模型参数值较小时，其导数值也会随之减小，从而达到减小模型容量、实现正则化的目的。

图 2-11 给出了参数向量为二维时利用L^2范数惩罚所能实现的约束效果，其中等值线为原代价函数$F(\boldsymbol{\beta})$的图像，圆周曲线为L^2范数惩罚项图像。L^2范数惩罚项处处可导，不会出现尖点，故难以获得稀疏参向量，但可通过减小模型参数值的方式实现正则化。

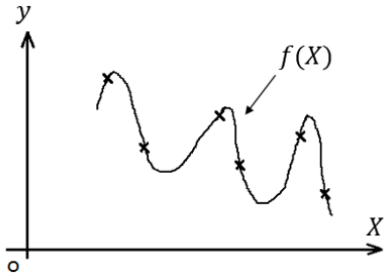


图 2-12 过拟合的回归模型示意图

从数值计算角度而言，若目标函数为二次函数，例如最小二乘估计中的目标函数：

$$F(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

则可直接算得解的具体表达式：

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

但若模型参数过多，则说明样本X由过多的特征所描述。若特征向量维数大于训练样本数量时，则矩阵$\boldsymbol{X}^T\boldsymbol{X}$不满秩。此时无法对其求逆，故难以直接计算获得目标函数的解。但若对目标函数添加L^2范数惩罚正则化项，则可将目标函数转化为以下形式：

$$F(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \frac{1}{2}\alpha\boldsymbol{\beta}^T\boldsymbol{\beta} \qquad (2\text{-}59)$$

得到其最优解的具体表达式为：

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X} + \alpha\boldsymbol{E})^{-1}\mathbf{X}^T\mathbf{y} \qquad (2\text{-}60)$$

其中$\boldsymbol{E}$为单位矩阵， $(\boldsymbol{X}^T\boldsymbol{X} + \alpha\boldsymbol{E})$为满秩矩阵，可直接通过求逆矩阵获得。

事实上，范数惩罚项可以是一般的L^p范数形式，即$\tau = \alpha\|\boldsymbol{\beta}\|_p^p$。其中$p$为正实数，对于$p$的不同取值，所获得的正则化效果也有所不同。 α则是一个适当的调和系数，当α较大时，模型参数都很小，此时模型学习能力较弱，可能会导致模型欠拟合；当α值较小时，相当于没有进行正则化，模型可能出现过拟合。因此，只有在α取值合适时，才能使得模型既能很好地拟合训练数据集，又不出现严重过拟合现象。

需要注意的是，模型偏置项一般不参与范数惩罚计算。这是因为确定偏置项所需样本量较少，故其对模型拟合能力的影响非常小，一般无需对其进行正则化处理。

2.4.2 样本增强

导致模型产生过拟合现象的原因是模型容量与训练样本数量不匹配，即由于训练样本数量较少从而导致模型的某些参数失去约束。因此，除了对模型容量进行约束之外，还可以通过增加训练样本数量的方式解决模型过拟合问题。样本增强便是一种基于扩充训练样本集正

则化策略。所谓**样本增强**就是通过适当方式从已有样本中产生一个或多个虚拟样本，以满足模型训练的需要。

显然，通过样本增强产生的虚拟样本需要满足一定的合理性，即既要与现有样本保持一定差异，又要服从与现有样本一致的总体概率分布。否则，使用这些虚拟样本不但不能减轻过拟合现象，反而会进一步降低模型的鲁棒性。在很多应用场合，对虚拟样本的合理性进行判断是一件非常困难的事情。然而，计算机视觉领域的处理对象是直观图像，可通过观察方式直观判断虚拟样本的合理性。因此，样本增强目前主要用于计算机视觉领域。

图像作在计算机视觉领域的处理对象，通常包含很多不改变原始信息表达的可变因素。例如一幅猫的图像，若改变其亮度或对其进行旋转，则图像中目标物体依旧是猫。故可通过改变图像中可变因素产生虚拟样本。最常见的样本增强方法包括旋转、翻转和加噪等。这些方法虽然较为简单，却能产生很多有效的虚拟样本，实现对训练样本集合的有效扩充。图 2-13 中的四幅图像分别表示原图以及通过旋转、垂直翻转、加噪音获得的三个虚拟样本。

（a）原始图像　　（b）旋转图像　　（c）垂直翻转图像　　（d）加噪音图像

图 2-13 通过样本增强获得虚拟样本

对于任意给定的一张数字图像X，若将其绕图像中心顺时针旋转θ度，则需将图像中心作为坐标原点建立一个坐标系。假设图像X某像素点A关于该坐标系的坐标为(x_A, y_A)，则其极坐标表示为$x_A = r\cos\alpha, y_A = r\sin\alpha$。其中$r$表示像素点$A$到坐标原点的距离，$\alpha$表示坐标原点与像素点$A$所在射线与极坐标轴之间的夹角大小。此时，将像素点$A$绕旋转中心逆时针旋转$\theta$度所得到像素点$A$新的坐标为：

$$\begin{cases} x'_A = r\cos(\alpha+\theta) = r\cos\alpha\cos\theta - r\sin\alpha\sin\theta = x_A\cos\theta - y_A\sin\theta \\ y'_A = r\sin(\alpha+\theta) = r\sin\alpha\cos\theta + r\cos\alpha\sin\theta = x_A\sin\theta + y_A\cos\theta \end{cases}$$

需将该图像中全部像素点按上式计算得到旋转后的坐标并移动至该位置即可得到旋转后的图像。值得注意的是，样本增强所得到的新样本类别应与原始样本保持一致。例如对字符 9 旋转 180 度得到样本类别为 6，这种样本增强方式则是不适用的。

与图像旋转不同，图像翻转生成的是与原图像X成轴对称的图像，具体可分为水平翻转和垂直翻转两种。水平翻转是指以图像的垂直中轴线为中心交换图像的左右两个部分。假设大小为$m \times n$的原图中某个像素点A的坐标为(x_A, y_A)，则进行水平翻转后像素点A所对应的新的坐标计算公式为：$x'_A = x_A, y'_A = m - y_A + 1$。垂直翻转是指以图像的水平中轴线为中心交换图像的上下两个部分，相应计算公式为$x'_A = n - x_A + 1, y'_A = y_A$。

图像加噪也是一种常用样本增强方法。其基本思想是对原图叠加一个微小的随机噪声，由此生成新样本。具体地说，对于给定的原始图像X，首先生成某种噪声ε，然后以简单叠加方式将噪声ε与原始图像进行合成，得到新图像X'，即$X' = X + \varepsilon$。

最近人们研究出一种基于生成对抗网络（GAN）的样本增强方法。该方法首先通过 GAN 模型的生成器生成新的样本，然后将生成的虚拟样本和训练集中实际图像样本随机输入判别器中，通过 GAN 模型的判别器判别输入样本是否为虚拟样本。最后，通过判别器的判别结果不断提升生成器的性能，使得判别器最终将无法判别输入样本是否为虚拟样本，此时所生成的虚拟样本即为所求。

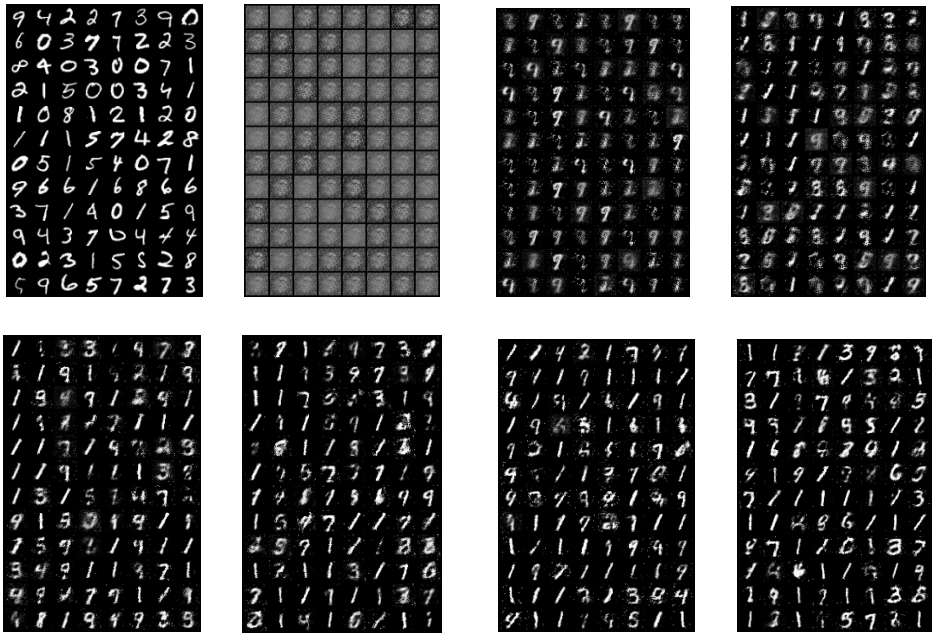


图 2-14 生成对抗网络生成效果提升过程示意图

图 2-14 表示 GAN 模型在训练过程中所生成的虚拟图像样本变化情况。其中第一幅图像为原始图像，其余为生成器所生成的虚拟图像。它们按照迭代次数的大小从左到右、从上到下进行排列。显然，随着生成器与判别器博弈，虚拟图像越来越接近真实图像，甚至能够以假乱真。图 2-20 表示使用训练好生成器生成的四张虚拟样本。

图 2-15 根据原图生成的虚拟样本

2.4.3 对抗训练

前述图像样本增强方法中，可对图像样本X加入少量噪声但不改变其类别就可得到新的虚拟样本X'。如果直接将虚拟样本X'作为测试样本而不是训练样本，则对于一些鲁棒性较弱的机器学习模型，可能会对虚拟样本X'做出错误分类。通常将这种与原始样本之间仅有少量差异却被错误分类的虚拟样本称为**对抗样本**。

对抗样本对机器学习模型具有很大的危害性，无论是传统机器学习模型还是目前常用的深度网络模型在对抗样本面前表现出一定的脆弱性。更加危险的是，有时同一对抗样本在不同模型上的测试结果出现同样的错误，也就是说对抗样本可能会同时威胁到多个模型。因此，必须尽可能找出对抗样本并对其进行训练，以提高模型的鲁棒性。

通常认为出现对抗样本的主要原因是由于模型在高维空间中作为一种线性模型而引起的误差累积放大。具体地说，对于线性模型$f(X)=X\boldsymbol{\beta}$，考虑对抗样本$X'=X+\varepsilon$和真实样本X在该模型下输出的差异：

$$f(X')-f(X)=\varepsilon_1\beta_1+\varepsilon_2\beta_2+\cdots+\varepsilon_k\beta_k \tag{2-61}$$

当特征向量维数k较大时，误差的线性累加会造成对抗样本在线性模型上输出与真实样本在其上输出误差变大，由此产生错误分类结果。

解决此类问题的直观方法是生成尽可能多对抗样本作为训练样本用于训练模型，纠正模型对对抗样本的错误分类，提高模型的鲁棒性。这种正则化策略通常称之为**对抗训练**。显然，对抗训练的关键在于如何生成用于训练的对抗样本。常用对抗样本生成方法有简单界约束限制域拟牛顿法、快速梯度符号法等。

简单界约束限制域拟牛顿法的基本思想是直接对优化目标添加对噪声ε的约束，使得添加了噪声ε的新样本被错误分类。具体来说，对于一个分类器f，该方法直接通过最优化如下目标生成对抗样本：

$$\min||\varepsilon||_2;\ \text{s.t.}\begin{cases} f(X+\varepsilon)=l \\ \text{label}(X+\varepsilon)=\text{label}(X)\neq l \end{cases}$$

其中$\text{label}(X)$为X的真实标注值，l为期望模型对生成样本的错误分类标记。

上述优化目标表示希望取得最小的噪声ε，使得模型将加噪后的生成样本分类为错误的类别l，并且保证生成样本$X'=X+\varepsilon$与原始样本X的区别不大。

为方便计算，可将上述优化问题的形式表示为：

$$\min[c|\varepsilon|+F(X+\varepsilon,l)];\ \text{s.t.}\quad \text{label}(X+\varepsilon)=\text{label}(X)\neq l \tag{2-62}$$

其中$c>0$，F为目标函数。

求解上述优化问题便可得到对抗样本。当上述目标函数为凸函数时，必定可以找到精确的对抗样本，即噪声ε最小的对抗样本。然而，实际问题的目标函数往往为非凸，此时只能找到近似对抗样本，即噪声ε并非最小值，但即便如此，所生成对抗样本与真实样本之间的差别也是非常小的。

简单界约束限制域拟牛顿法法虽然可以生成对抗样本，但其效率低下。事实上，可根据对抗样本产生原因构造对抗样本生成算法。由此可得用于快速生成抗样本的快速梯度符号法。该方法认为机器学习模型在高维空间中是一个高度线性的模型。此时，对样本添加一个较小的噪声，就通过高维线性模型对噪声的累加改变模型输出结果，得到对抗样本。

具体地说，设模型参数向量为$\boldsymbol{\beta}=(\beta_1,\beta_2,\ldots,\beta_k)^T$，样本输入为$X$。若想得到一个对抗样本$X'$，可通过对样本$X$添加如下最优扰动来实现：

$$\varepsilon=\gamma\text{sign}\left(\nabla_X F(\boldsymbol{\beta},X,y)\right) \tag{2-63}$$

即生成样本为：

$$X'=X+\ \gamma\text{sign}\left(\nabla_X F(\boldsymbol{\beta},X,y)\right) \tag{2-64}$$

上式中，$F(\boldsymbol{\beta},X,y)$表示模型$f$参数向量为$\boldsymbol{\beta}$时将$X$分类为$y$的累计损失。在生成对抗样本时将$X$和$y$视为变量。$\text{sign}\left(\nabla_X F(\boldsymbol{\beta},X,y)\right)$为模型目标函数对$X$的梯度方向，$\gamma$为噪声在该方向上的偏移参数，通常由人为给定。

快速梯度符号法的基本思想是在目标函数的梯度方向上对原始样本X添加扰动使得目标函数值增大，由此提升样本错误分类的概率。当模型为近似高维线性模型时，一个很小的扰动就会由于扰动的累加导致模型输出的发生较大的改变，由此生成所需的对抗样本。需要注意的是，快速梯度符号法在生成对抗样本时并未指明将对抗样本进行错分的具体类别，故由该方法生成对抗样本的错分类别是不确定的，即为不定向对抗样本。事实上，也可使用快速梯度符号法生成定向对抗样本。

在使用上述方法或其他方法生成对抗样本之后，可将这些对抗样本添加到训练样本集当中构成新的训练样本集重新训练模型，便可完成对抗训练。不难看出，对抗训练在本质上是一种特殊的数据增强正则化方法，只是用于扩充训练样本集的虚拟样本较为特殊，通过添加这些对模型鲁棒性有较大影响的样本到训练集中以尽可能提升模型的鲁棒性。

2.5 习 题

（1）现有一组某市的房价与其位置数据如表 2-12 所示，其中D表示房屋到市中心的直线距离，单位为千米，R表示房屋单价，单位为元/平方米。试根据以下数据使用最小二乘估计确定房价于其位置之间的大致关系。

表 2-12 房价与其位置数据表

序号	1	2	3	4	5	6	7	8	9
D	4.2	7.1	6.3	1.1	0.2	4.0	3.5	8	2.3
R	8600	6100	6700	12000	14200	8500	8900	6200	11200

（2）假设某学校男生的身高服从正态分布$N(\mu,\sigma^2)$，现从全校所有男生中随机采样测量得到身高数据如表 2-13 所示，试通过表中数据使用最大似然估计法估计μ和σ^2的取值。

表 2-13 身高数据表

序号	1	2	3	4	5	6	7	8
身高（cm）	167	175	163	169	174	187	168	176

（3）假设某学校男生的身高服从正态分布$N(\mu,\sigma^2)$，上一次测试时得到身高均值的估计值为172cm，方差为36，故在本次测试前，以0.7的概率相信该校男生身高服从$N(172,\ 36)$，试根据表 2-13 中数据和最大后验估计法确定μ和σ^2的估计值。

（4）试用梯度下降算法求解无约束非线性规划问题：

$$\min f(X) = (x_1 - 2)^4 + (x_1 - 2x_2)^2$$

其中$X = (x_1, x_2)^T$，要求选取初始点$X^0 = (0,3)^T$，终止误差$\varepsilon = 0.1$。

（5）若要使用表 2-12 中数据构建一个用于预测房屋价格与房屋到市区距离之间关系的线性模型，其中模型优化过程使用梯度下降算法，试取任意初始点开始迭代，步长取 0.05，计算前两次迭代的结果。

（6）与共轭梯度法相比较，梯度下降法有何缺陷？共轭梯度法为何能避免这种缺陷？

（7）利用共轭梯度算法求解无约束非线性规划问题：

$$\min f(X) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

其中$X = (x_1, x_2)^T$，取迭代起始点为$X^0 = (1,1)^T$。

（8）若要使用表 2-12 中数据构建一个用于预测房屋价格与房屋到市区距离之间关系的线性模型，其中模型优化过程使用共轭梯度法，试取任意初始点开始迭代，计算前两次迭代的结果。

（9）使用牛顿法求解无约束非线性规划问题：

$$\min f(X) = (x_1 - x_2)^3 + (x_1 + 3x_2)^2$$

其中$X = (x_1, x_2)^T$，取迭代起始点为$X^0 = (1,2)^T$。

（10）牛顿法存在哪些缺陷？拟牛顿法为何能克服这些缺陷？

（11）使用拟牛顿法求解无约束非线性规划问题：

$$\min f(X) = (4 - x_2)^3 + (x_1 + 4x_2)^2$$

其中$X = (x_1, x_2)^T$，取迭代起始点为$X^0 = (2,1)^T$。

（12）证明当损失函数是对数损失函数时，经验风险最小化等价于极大似然估计。

（13）与梯度下降方法相比，随机梯度方法为何能降低算法的时间复杂度？

（14）小批量随机梯度下降法与随机梯度下降法有何区别？这样设计小批量随机梯度下降法的原因是什么？

（15）证明：设$P(Y|\theta)$为观测数据的似然函数，$\theta^{(i)}(i = 1,2,\cdots)$为EM算法得到的参数估计序列，$P(Y|\theta^{(i)})(i = 1,2,\cdots)$为对应的似然函数序列，则$P(Y|\theta^{(i)})$是单调递增的。

（16）蒙特卡洛方法的理论基础是什么？如何使用蒙特卡洛方法估计圆周率的取值？马尔可夫链蒙特卡罗方法有哪些具体应用？

（17）模型的正则化方法有哪些？它们分别是从何种角度出发对模型进行正则化的？

（18）范数惩罚正则化中，使用L^1范数惩罚可以达到什么样的约束效果？使用L^2范数惩罚又能达到何种约束效果？能够达到这些约束效果的原因是什么？

（19）对于图像X，假设在以图像中心点为原点建立的坐标系中，某个像素点的坐标为(x, y)，试求将该图像顺时针旋转θ度后该像素点所对应的新的坐标。

（20）对抗样本的存在会对机器学习模型造成怎样的危害？对抗训练方法为何能提升模型的鲁棒性？