第 3 单元　线性模型

支撑的课程目标

1. 能够基于智能信息处理的基本理论和技术，识别和理解数据处理与分析等问题的相关特性.
2. 能够运用智能信息处理的相关原理和专业知识，设计实验方案，为解决数据处理与分析等问题提供支持.

基本要求

1. 应用线性回归模型和逻辑回归模型，解决数据分析领域的回归问题.
2. 应用损失函数、优化器和模型可视化，分析回归模型的问题.

教学重点与难点

重点：　多元线性回归；逻辑回归.

难点：　归回模型的学习原理.

教学过程设计

新课导入、知识讲授、教学目标达成考核、总结.

教学过程设计

本单元教学通过“互动、开放”的课堂形式，采用探究式学习、问题导入的教学方法，激发学生的学习兴趣，促成课程目标的达成.

教学学时

6 学时.

一、导入新课（5 分钟）

回归是指这样一类问题：通过统计分析一组随机变量 $x_1, \cdots, x_n$ 与另一组随机变量 $y_1, \cdots, y_n$ 之间的关系，得到一个可靠的模型，使得对于给定的 $x =$

$\{x_1, \cdots, x_n\}$，可以利用这个模型对 $y = \{y_1, \cdots, y_n\}$ 进行预测。在这里，随机变量 $x_1, \cdots, x_n$ 被称为自变量，随机变量 $y_1, \cdots, y_n$ 被称为因变量。

不失一般性，我们在本单元讨论回归问题的时候，总是假设因变量只有一个。这是因为我们假设各因变量之间是相互独立的，因而多个因变量的问题可以分解成多个回归问题加以解决。

形式化地，在回归中我们有一些数据样本 $\{(x_i, y_i)\}_{i=1}^{N}$，通过对这些样本进行统计分析，我们获得一个预测模型 $f(\cdot)$，使得对于测试数据 $x = \{x_1, \cdots, x_n\}$，可以得到一个较好的预测值：

$$y = f(x)$$

二、新课讲授 (240 分钟)

本单元要点:

* 回归的线性模型（一元线性回归和多元线性回归）

* 分类的线性模型（logistic 回归和 softmax 回归）

1. 线性回归

1.1 一元线性回归

线性回归模型是指 $f(\cdot)$ 采用线性组合形式的回归模型，在线性回归问题中，因变量和自变量之间是线性关系的。对于因变量 x，我们乘以权重系数 w，取 y 为因变量的线性表示：

$$y = f(x) = wx + b$$

其中 b 为常数项。可以看到 w 和 b 决定了回归模型 $f(\cdot)$ 的行为。在这里我们介绍最小二乘法求解线性回归中参数估计的问题。

受教育程度与年度收入之间的关系预测：根据社会经验和统计数据分析，收入会随着受教育时间的增多而增多。线性回归试图以受教育年限为输入 x_i，以年度收入 y_i 为输出，学习得到 $y = wx_i + b$，使得 $y \cong y_i$。

线性回归的关键问题在于确定参数 w 和 b，使得拟合输出 y 与真实输出 y_i 尽可能相近。在回归任务中我们通常使用均方误差来度量预测值与标签之间的损失，所以回归任务的优化目标就是使得拟合输出和真实输出之间的均方误差

最小化（误差函数 $L(w,b)$ 最小化）：

$$\begin{aligned} w,b &= arg\min_{w,b} L(w,b) \\ &= arg\min_{w,b} \frac{1}{N}\sum_{i=1}^{N}(y-y_i)^2 \\ &= arg\min_{w,b} \frac{1}{N}\sum_{i=1}^{N}(wx_i+b-y_i)^2 \end{aligned} \tag{1}$$

为求得误差函数最小时的参数 w 和 b，分别对 w 和 b 求一阶导数并令其等于 0。对 w 求导的推导过程如下：

$$\begin{aligned} \frac{\partial L(w,b)}{\partial w} &= \frac{\partial}{\partial w}\left[\frac{1}{N}\sum_{i=1}^{N}(wx_i+b-y_i)^2\right] \\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{\partial}{\partial w}\left[(y_i-wx_i-b)^2\right] \\ &= \frac{2}{N}\sum_{i=1}^{N}\left[(y_i-wx_i-b)\cdot(-x_i)\right] \\ &= \frac{2}{N}\sum_{i=1}^{N}(wx_i^2-y_ix_i+bx_i) \\ &= 2\cdot\left(w\frac{1}{N}\sum_{i=1}^{N}x_i^2-\frac{1}{N}\sum_{i=1}^{N}y_ix_i+b\frac{1}{N}\sum_{i=1}^{N}x_i\right) \end{aligned} \tag{2}$$

对 b 求导的推导过程如下：

$$\begin{aligned}\frac{\partial L(w,b)}{\partial b} &= \frac{\partial}{\partial b}\left[\frac{1}{N}\sum_{i=1}^{N}(wx_i+b-y_i)^2\right] \\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{\partial}{\partial b}\left[(y_i-wx_i-b)^2\right] \\ &= \frac{2}{N}\sum_{i=1}^{N}\left[(y_i-wx_i-b)\cdot(-1)\right] \\ &= \frac{2}{N}\sum_{i=1}^{N}(wx_i-y_i+b) \\ &= 2\cdot\left(w\frac{1}{N}\sum_{i=1}^{N}x_i-\frac{1}{N}\sum_{i=1}^{N}y_i+\frac{1}{N}\sum_{i=1}^{N}b\right)\end{aligned} \tag{3}$$

令

$$\begin{aligned}\overline{x} &= \frac{1}{N}\sum_{i=1}^{N}x_i \\ \overline{y} &= \frac{1}{N}\sum_{i=1}^{N}y_i \\ s^2 &= \frac{1}{N}\sum_{i=1}^{N}x_i^2 \\ \rho &= \frac{1}{N}\sum_{i=1}^{N}y_ix_i\end{aligned} \tag{4}$$

令 $L(w,b)$ 关于 w 和 b 的导数等于 0，得到

$$\begin{aligned}\frac{\partial L(w,b)}{\partial w} &= 0 \Rightarrow ws^2+b\overline{x}-\rho=0 \\ \frac{\partial L(w,b)}{\partial b} &= 0 \Rightarrow w\overline{x}+b-\overline{y}=0\end{aligned} \tag{5}$$

当导数为 0 时，可以求得损失函数取最小值时的 w 和 b 求解二元一次方程，

得到

$$
\begin{aligned}
w &= \frac{\rho - \overline{yx}}{s^2 - \overline{x}^2} \\
b &= \overline{y} - w\overline{x}
\end{aligned}
\tag{6}
$$

1.2 多元线性回归

在多元线性回归中，对于第 j 个因变量 x_j，我们乘以权重系数 w_j，取 y 为因变量的线性组合：

$$
y = f(\mathbf{x}) = w_1 x_1 + \cdots + w_{M-1} x_{M-1} + b
$$

其中 b 为常数项。若令 $\mathbf{w} = (w_1, \cdots, w_{M-1})^T$，$\mathbf{x} = (x_1, \cdots, x_{M-1})^T$，则上式可以写成向量形式：

$$
y = f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b
$$

可以看到 $\mathbf{w}$ 和 b 决定了回归模型 $f(\cdot)$ 的行为。在这里我们介绍最小二乘法求解线性回归中参数估计的问题。

二手房房价预测：影响二手房房屋单价的主要因素包括面积、户型、地段、房龄等因素。线性回归试图以面积、户型、地段、房龄等影响因素作为输入 $\mathbf{x}_i = (x_1^i, \cdots, x_{M-1}^i)$，以房屋单价 y_i 作为输出，$i \in [1, \cdots, N]$，学习得到 $y = \mathbf{w}^T \mathbf{x} + b$，使得 $y \cong y_i$。

线性回归的关键问题在于确定参数 $\mathbf{w}$ 和 b，使得拟合输出 y 与真实输出 y_i 尽可能相近。在回归任务中我们通常使用均方误差来度量预测值与标签之间的损失，所以回归任务的优化目标就是使得拟合输出和真实输出之间的均方误差最小化

$$
\begin{aligned}
\mathbf{w}, b &= arg\min_{\mathbf{w},\mathbf{b}} L(\mathbf{w}, b) \\
&= arg\min_{\mathbf{w},b} \sum_{i=1}^{N} (y - y_i)^2 \\
&= arg\min_{\mathbf{w},b} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x} + b - y_i)^2
\end{aligned}
\tag{7}
$$

为了便于计算参数，将参数 b 合并到向量 $\mathbf{w}$，即令 $\mathbf{w} = (w_0, w_1, \cdots, w_{M-1})^T$，$\mathbf{x} = (1, x_1, \cdots, x_{M-1})^T$，其中 $b = w_0$。则 y 可以重写为：

$$y = f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$$

对于数据集 $\mathbf{x}_i$，y_i，$i \in [1, \cdots, N]$，回归模型的损失函数 $L(\mathbf{w})$ 表示为

$$\begin{aligned} L(\mathbf{w}) &= \sum_{i=1}^{N}(y_i - y)^2 \\ &= \sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 \\ &= \sum_{i=1}^{N}(y_i - \mathbf{x}_i^T\mathbf{w})^2 \\ &= (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) \\ &= \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} - \mathbf{w}^T\mathbf{X}^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} \end{aligned} \tag{8}$$

其中

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \cdots \\ \mathbf{x}_N^T \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1(M-1)} \\ 1 & x_{21} & x_{22} & \cdots & x_{2(M-1)} \\ & & \cdots & & \\ 1 & x_{N1} & x_{N2} & \cdots & x_{N(M-1)} \end{bmatrix} \tag{9}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{bmatrix}$$

因此，回归模型的优化函数可以改写为

$$\begin{aligned} \mathbf{w} &= arg\min_{\mathbf{w}} L(\mathbf{w}) \\ &= arg\min_{\mathbf{w}} (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) \\ &= arg\min_{\mathbf{w}} \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} - \mathbf{w}^T\mathbf{X}^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} \end{aligned} \tag{10}$$

根据矩阵微分公式：

$$\begin{aligned}\frac{\partial \mathbf{a}^T\mathbf{x}}{\partial \mathbf{x}} &= \frac{\partial \mathbf{x}^T\mathbf{a}}{\partial \mathbf{x}} = \mathbf{a} \\ \frac{\partial \mathbf{x}^T\mathbf{A}\mathbf{x}}{\partial \mathbf{x}} &= (\mathbf{A}+\mathbf{A}^T)\mathbf{x}\end{aligned} \tag{11}$$

$\mathbf{w}$ 求导的推导过程如下：

$$\begin{aligned}\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}}\left[\mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\mathbf{w} - \mathbf{w}^T\mathbf{X}^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}\right] \\ &= 0 - \mathbf{X}^T\mathbf{y} - \mathbf{X}^T\mathbf{y} + (\mathbf{X}^T\mathbf{X} + \mathbf{X}^T\mathbf{X})\mathbf{w} \\ &= 2\mathbf{X}^T\mathbf{X}w - 2\mathbf{X}^T\mathbf{y} \\ &= 2\mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y})\end{aligned} \tag{12}$$

当矩阵 $\mathbf{X}^T\mathbf{X}$ 为满秩矩阵或正定矩阵时，令 $\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = 0$，可解得参数 $\mathbf{w}$ 为

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{13}$$

2. 线性分类

在线性回归中，我们假设随机变量 $x_1, \cdots, x_n$ 与 y 之间的关系是线性的。但在实际中，我们通常会遇到非线性关系。这个时候，我们可以使用一个非线性变换 $g(\cdot)$，对线性回归模型的结果 $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ 进行变换得到输出 y，所以逻辑回归中我们实际上要拟合的是 $x_1, \cdots, x_n$ 与 $g(\mathbf{w}^T\mathbf{x})$ 的关系（在线性回归中拟合的是 $x_1, \cdots, x_n$ 与 $\mathbf{w}^T\mathbf{x}$ 之间的关系）：

$$y = g(f(\mathbf{x}))$$

其中 $f(\cdot)$ 仍然为：

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$$

这样的回归模型称为广义线性回归模型。广义线性回归模型使用非常广泛。

2.1 贝努力分布

2.1.1 二值贝努力分布伯努利分布（Bernoulli distribution）：单次随机实验，结果有两种可能：成功和失败。例如利用 x 描述抛一枚硬币的结果，$x = 1$ 表示正面

朝上，$x=0$ 表示背面朝上。假设这枚硬币受损了，正面朝上的概率与背面朝上的概率不一定一样。

假设 $x \in \{0,1\}$ 是一个二值随机变量（binary random variable），$x=1$ 的概率为 μ，即 $p(x=1|\mu)=\mu$；$x=0$ 的概率为 $1-\mu$，即 $p(x=0|\mu)=1-\mu$，则称 x 服从伯努利分布。x 的概率分布函数（probability distribution）定义为：

$$Bern(x|\mu)=\mu^x(1-\mu)^{1-x} \tag{14}$$

假设获取随机变量 x 的一组观测值 $D=\{x_1,x_2,\cdots,x_N\}$，并且这些观测值是独立采样自概率分布 $p(x|\mu)$，则构建似然函数（μ 的函数）如下：

$$p(D|\mu)=\prod_{n=1}^{N} lnp(x_n|\mu)=\prod_{n=1}^{N}\mu^{x_n}(1-\mu)^{1-x_n} \tag{15}$$

由于对数函数 ln 是增函数，最大化似然函数 $p(D|\mu)$ 等价于最大化似然函数 $p(D|\mu)$ 的对数。贝努力分布的对数似然函数有下式给出

$$p(D|\mu)=\sum_{n=1}^{N} lnp(x_n|\mu)=\sum_{n=1}^{N}\{x_n ln\mu+(1-x_n)ln(1-\mu)\} \tag{16}$$

2.1.1 多项贝努力分布

多项伯努利分布（multinoulli distribution）：单次随机实验，结果有 K 种可能。假设一个 K 维向量 $x=\{x_1,\cdots,x_K\}$，其中 $x_k=1$，那么其它的变量都等于零。例如，一个随机变量取 6 种状态，且恰好观测到 $x_3=1$，那么有

$$\vec{x}=(0,0,1,0,0,0)^T \tag{17}$$

假设 $x=(x_1,x_2,\cdots,x_K)$ 是一个随机向量（random vector），$x_k \in \{0,1\}$ 是二值随机变量，$x_k=1$ 的概率为 μ_k（当 $x_k=1$ 时，其它变量 $x_j=0, j\neq 0$），则称 x 服从多项伯努利分布。x 的概率分布函数（probability distribution）定义为：

$$p(\mathbf{x}|\mu)=\prod_{k=1}^{N}\mu_k^{x_k} \tag{18}$$

假设获取随机变量 x 的一组观测值 $D = \{x_1, x_2, \cdots, x_N\}$，并且这些观测值是独立采样自概率分布 $p(x|\mu)$，则构建似然函数（μ 的函数）如下：

$$p(D|\mu) = \prod_{n=1}^{N} lnp(\mathbf{x}_n|\mu) = \prod_{n=1}^{N}\prod_{k=1}^{N} \mu_k^{x_k} \tag{19}$$

2.2 logistic 回归

在二元分类任务中，我们的目标是通过这样一个分离超平面 $a(\mathbf{x})) = \mathbf{w}^T\mathbf{x}$ 获得目标分类 y，使得 y 可表示为以下阶跃函数：

$$y = \begin{cases} 0 & f(\mathbf{x}) < 0 \\ 1 & f(\mathbf{x}) > 0 \end{cases}$$

但是在分类问题中，由于 y 取离散值，这个阶跃判别函数是不可导的。不可导的性质使得许多数学方法不能使用。我们考虑使用一个函数 $\sigma(\cdot)$ 来近似这个离散的阶跃函数，通常可以使用 logistic 函数或 tanh 函数。

令

$$\sigma(a) = \frac{1}{1 + exp(-a)}$$

使用 logistic 函数代替阶跃函数，使得分类函数可以表示为：

$$y = \sigma(a(\mathbf{x})) = \frac{1}{1 + exp(-\mathbf{w}^T\mathbf{x})} = p \tag{20}$$

这样就可以把离散取值的分类问题近似地表示为连续取值的回归问题；这样的回归模型称为 logistic 回归模型。

由于 logistic 函数的取值范围在 $[0, \cdots, 1]$ 之间，所以 $\sigma(a(\mathbf{x}))$ 可以看做概率函数。将 y 为标签 1 的概率记作 $p(y = 1|\mathbf{x})$ 概率，可以定义如下的条件概率：

$$\begin{aligned} P(y = 1|\mathbf{x}) &= p \\ P(y = 0|\mathbf{x}) &= 1 - p \end{aligned} \tag{21}$$

因为根据全概率公式有 $p(y = 1|\mathbf{x}) + p(y = 0|\mathbf{x}) = 1$。根据上述两个公式，可以写出 $p(y|\mathbf{x})$ 的概率公式 (贝努力公式)：

$$P(y|\mathbf{x}) = p^y(1 - p)^{1-y}$$

解释下这个函数的含义，我们采集到了一个样本 $(\mathbf{x}_i, y_i)$。对这个样本，它的标签是 y_i 的概率是 $p^y(1-p)^{1-y}$。(当 $y=1$，结果是 p；当 $y=0$，结果是 $1-p$)。

如果我们采集到了一组数据一共N个,$\{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_N, \mathbf{y}_N)\}$,假设 $\mathbf{x}_1, \cdots, \mathbf{x}_N$ 相互独立，则似然函数（联合概率密度函数）为

$$\begin{aligned} L(\mathbf{x}_1, \cdots, \mathbf{x}_N; \mathbf{w}) &= P(y_1|\mathbf{x}_1; \mathbf{w}) \cdots P(y_N|\mathbf{x}_N; \mathbf{w}) \\ &= \prod_{i=1}^{N} P(y_i|\mathbf{x}_i; \mathbf{w}) \\ &= \prod_{i=1}^{N} p^{y_i}(1-p)^{1-y_i} \end{aligned} \tag{22}$$

我们希望的是联合概率越大越好。首先，我们对联合概率引入 ln 函数，因为 ln 运算并不会影响函数本身的单调性。则有:

$$\begin{aligned} ln(L(\mathbf{x}_1, \cdots, \mathbf{x}_N; \mathbf{w})) &= ln \prod_{i=1}^{N} p^{y_i}(1-p)^{1-y_i} \\ &= \sum_{i=1}^{N} y_i ln(p) + (1-y_i) ln(1-p) \end{aligned} \tag{23}$$

这样一来，问题就变成了以对数似然函数为目标函数的最优化问题，这里可以使用梯度上升法求使得 $ln(L(\mathbf{x}_1, \cdots, \mathbf{x}_N; \mathbf{w}))$ 最大时的 $\mathbf{w}$ 值。但是这里我们令:

$$\begin{aligned} J(\mathbf{w}) &= -\frac{1}{N} ln(L(\mathbf{x}_1, \cdots, \mathbf{x}_N; \mathbf{w})) \\ &= -\frac{1}{N} \sum_{i=1}^{N} y_i ln(p) + (1-y_i) ln(1-p) \end{aligned} \tag{24}$$

w 求导的推导过程如下：

$$\begin{aligned}\nabla J(w)=\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}}\left[-\frac{1}{N}\sum_{i=1}^{N} y_i ln(p)+(1-y_i)ln(1-p)\right] \\ &= -\frac{1}{N}\sum_{i=1}^{N}\frac{\partial}{\partial \mathbf{w}}\left[y_i ln(p)+(1-y_i)ln(1-p)\right] \\ &= -\frac{1}{N}\sum_{i=1}^{N}\left[y_i\frac{\partial}{\partial \mathbf{w}}ln(p)+(1-y_i)\frac{\partial}{\partial \mathbf{w}}ln(1-p)\right] \\ &= -\frac{1}{N}\sum_{i=1}^{N}\left[y_i\frac{1}{p}\frac{\partial p}{\partial \mathbf{w}}+(1-y_i)\frac{1}{1-p}\frac{\partial 1-p}{\partial \mathbf{w}}\right] \\ &= -\frac{1}{N}\sum_{i=1}^{N}\left[y_i\frac{1}{p}p(1-p)\mathbf{x}+(1-y_i)\frac{1}{1-p}(-p)(1-p)\mathbf{x}\right] \\ &= -\frac{1}{N}\sum_{i=1}^{N}(y_i-p)\mathbf{x}\end{aligned} \tag{25}$$

现在我们已经解出了损失函数 $J(w)$ 在任意 w 处的梯度 $\nabla J(w)$，可是我们怎么算出来 w 呢？回到之前的问题，我们现在要求损失函数取最小值时候的 w 的值：

可以用梯度下降法 (Gradient Descent) 来解决这个问题。核心思想就是先随便初始化一个 w_0，然后给定一个步长 η，通过不断地修改 $w_{t+1}<w_t$，从而最后靠近到达取得最小值的点，即不断进行下面的迭代过程，直到达到指定次数，或者梯度等于 0 为止。

$$w_{t+1}=w_t-\eta\nabla J(w)$$

2.3 softmax 回归

在多分类任务中，我们的目标是通过这样一个分离超平面 $a(\mathbf{x}))=\mathbf{w}^T\mathbf{x}$ 获得目标分类 y，使得 y 可以用 $1-of-K$ 方案来表示，即 y 是一个二值向量（除了某个元素 $y_k=1$，其余的元素均等于零）。

在多分类问题中，$1-of-K$ 方案可以使用 softmax 函数近似，其表达式为

$$\sigma(a_k)=\frac{exp(a_k)}{\sum_j exp(a_j)}$$

利用 softmax 函数可以将分类函数表示为：

$$p(C_k|\mathbf{x}) = y_k(\mathbf{x}) = \sigma(a_k) = \frac{exp(a_k)}{\sum_j exp(a_j)} \tag{26}$$

其中' 激活值'$a_k(\mathbf{x}) = \mathbf{w}_k^T\mathbf{x}$。

因此，下面需要确定模型的参数 $\{\mathbf{w}_k\}$。首先，计算 y_k 关于 a_j 的导数由下式给出：

$$\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j) \tag{27}$$

上式计算如下：

$$\begin{aligned} \frac{\partial y_k}{\partial a_j} &= \frac{e^{a_k}}{\sum_i e^{a_i}} - \left(\frac{e^{a_k}}{\sum_i e^{a_i}}\right)^2 = y_k(1 - y_k) \\ \frac{\partial y_k}{\partial a_j} &= -\frac{e^{a_k}e^{a_j}}{(\sum_i e^{a_i})^2} = -y_k y_j \quad j \neq k \end{aligned} \tag{28}$$

综合上述两式，得到式 (27)。下面计算 $\{\mathbf{w}_k\}$ 似然函数。

假设我们采集到了一组 N 个数据，$\{(\mathbf{x}_1, \mathbf{t}_1), \cdots, (\mathbf{x}_N, \mathbf{t}_N)\}$，假设 $\mathbf{x}_1, \cdots, \mathbf{x}_N$ 相互独立，$\mathbf{y}_n$ 为输入 $\mathbf{x}_n$ 时获得的每个类的条件概率，则似然函数（联合概率密度函数）为

$$\begin{aligned} L(\mathbf{x}_1, \cdots, \mathbf{x}_N; \mathbf{w}_1, \cdots, \mathbf{w}_K) &= P(\mathbf{t}_1|\mathbf{x}_1) \cdots P(\mathbf{t}_N|\mathbf{x}_N) \\ &= \prod_{n=1}^{N} P(\mathbf{t}_n|\mathbf{x}_n) \\ &= \prod_{n=1}^{N}\prod_{k=1}^{K} y_{nk}^{t_{nk}} \end{aligned} \tag{29}$$

其中 $y_{nk} = y_k(\mathbf{x}_n)$，$t_{nk}$ 表示 $1ofK$ 形式的向量 $\mathbf{t}_n$ 的第 k 个分量。

我们取似然函数的负对数作为损失函数

$$\begin{aligned} J(\mathbf{w}_1, \cdots, \mathbf{w}_K) &= -lnL(\mathbf{x}_1, \cdots, \mathbf{x}_N; \mathbf{w}_1, \cdots, \mathbf{w}_K) \\ &= -\sum_{n=1}^{N} ln \prod_{k=1}^{K} y_{nk}^{t_{nk}} \\ &= -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk} \end{aligned} \tag{30}$$

其中 $l(\mathbf{y}_n, \mathbf{t}_n) = -\sum_{k=1}^{K} t_{nk} \ln y_{nk}$ 称为交叉熵损失函数。则 $J(\mathbf{w}_1, \cdots, \mathbf{w}_K)$ 关于 w_j 的梯度可以计算为

$$
\begin{aligned}
\nabla J(\mathbf{w}_1, \cdots, \mathbf{w}_K) &= \frac{\partial J(\mathbf{w}_1, \cdots, \mathbf{w}_K)}{\partial \mathbf{w}_j} \\
&= -\frac{\partial}{\partial \mathbf{w}_j} \left[\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_{nk} \right] \\
&= -\sum_{n=1}^{N} \sum_{k=1}^{K} \frac{\partial}{\partial \mathbf{w}_j} \left[t_{nk} \ln y_{nk} \right] \\
&= -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \frac{\partial ln\, y_{nk}}{\partial y_{nk}} \frac{\partial y_{nk}}{\partial a_{nj}} \frac{\partial a_{nj}}{\partial \mathbf{w}_j} \\
&= -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \frac{1}{\partial y_{nk}} y_{nk} (I_{kj} - y_{nj}) \mathbf{x}_n \\
&= -\sum_{n=1}^{N} \sum_{k=1}^{K} (t_{nk} I_{kj} - t_{nk} y_{nj}) \mathbf{x}_n \\
&= -\sum_{n=1}^{N} (t_{nj} - y_{nj}) \mathbf{x}_n \\
&= \sum_{n=1}^{N} (y_{nj} - t_{nj}) \mathbf{x}_n
\end{aligned}
\tag{31}
$$

其中 $\sum_{k=1}^{K} t_{nk} = 1$。

现在我们已经解出了损失函数 J 在 $\mathbf{w}_j$ 处的梯度 ∇J，可以用梯度下降法 (Gradient Descent) 来更新参数 $\mathbf{w}_j, \quad j = 1, \cdots, K$。核心思想就是先随便初始化一个 $\mathbf{w}_j^0$，然后给定一个步长 η，通过不断地修改 $\mathbf{w}_j^{m+1} < \mathbf{w}_j^m$，从而最后靠近到达取得最小值的点，即不断进行下面的迭代过程，直到达到指定次数，或者梯度等于 0 为止。

$$
\mathbf{w}_j^{m+1} = \mathbf{w}_j^m - \eta \frac{\partial J(\mathbf{w}_1, \cdots, \mathbf{w}_K)}{\partial \mathbf{w}_j}
$$

三、教学目标考核（50 分钟）

讨论：

1. 什么是一元线性回归？如何建立一元线性回归模型？

2. 什么是多元线性回归？如何建立多元线性回归模型？

3. 什么是逻辑回归？如何建立逻辑回归模型？

四、总结（5 分钟）

Logistic 回归是深度学习中最基础的非线性模型之一。作为铺垫，在介绍 Logistic 回归以前，首先介绍了线性回归。线性回归的预测目标是连续变量，而 Logistic 回归的预测目标是二元变量。为了应对这一差异，Logistic 回归在线性回归的基础上加入了 Sigmoid 激活函数。