

# 机器学习

李成龙

安徽大学人工智能学院

“多模态认知计算”安徽省重点实验室

合肥综合性国家科学中心人工智能研究院

- 什么是机器学习
- 机器如何学习
- 如何让机器学习的更好
- 为什么机器能学习

- 机器如何学习

- 有监督学习

- 感知机
    - 支持向量机
    - 朴素贝叶斯分类
    - 决策树
    - 集成学习（Bagging算法与随机森林、Boosting算法）
    - 线性回归
    - 逻辑回归
    - Softmax回归
    - 神经网络与深度学习

- 无监督学习

- 聚类
    - 主成分分析

# 本节目录



- 线性回归
- 逻辑回归
- Softmax回归

# 本节目录



安徽大學  
ANHUI UNIVERSITY



- 线性回归
- 逻辑回归
- Softmax回归

- 定义

- 回归定义：通过带标签样本训练构造适当模型并通过该模型算出新样本的预测值
- 线性回归：基于线性模型的回归学习任务通常称之为线性回归，相应的线性模型称为线性回归模型
- 对于任意给定的样本 $\mathbf{x}$ ，线性回归的初始模型表示为：

$$f(\mathbf{x}) = w_1x_1 + w_2x_2 + \cdots + w_mx_m$$

其中 $\mathbf{w} = (w_1, w_2, \dots, w_m)^T$ 为参数向量

## • 模型求解

- 给定训练样本 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$ , 使用最小二乘法, 即基于均方误差最小化进行模型求解:

$$\mathbf{w}^* = \operatorname{argmin} \sum_{i=1}^m (f(\mathbf{x}_i) - y_i)^2 = \operatorname{argmin} \sum_{i=1}^m (\mathbf{w}^T \mathbf{x}_i - y_i)^2$$

- 则:

$$J(\mathbf{w}) = \sum_{i=1}^n (\mathbf{w}^T \mathbf{x}_i - y_i)^2 = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$$

其中 $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)^T = (x_{ij})_{m \times n}$ ,  $\mathbf{y} = (y_1, y_2, \dots, y_m)^T$

- 令 $J(\mathbf{w})$ 对参数向量 $\mathbf{w}$ 各分量的偏导数为0, 即:

$$\frac{\partial J}{\partial \mathbf{w}} = \mathbf{X}^T (\mathbf{y} - \mathbf{X}\mathbf{w}) = 0$$

- 则由 $\mathbf{X}^T (\mathbf{y} - \mathbf{X}\mathbf{w}) = 0$ 解得:

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

- 例题：**某企业某商品月广告费用与月销售量数据如表所示，试通过线性回归模型分析预测这两组数据之间的关系

表 月广告费与月销售量数据

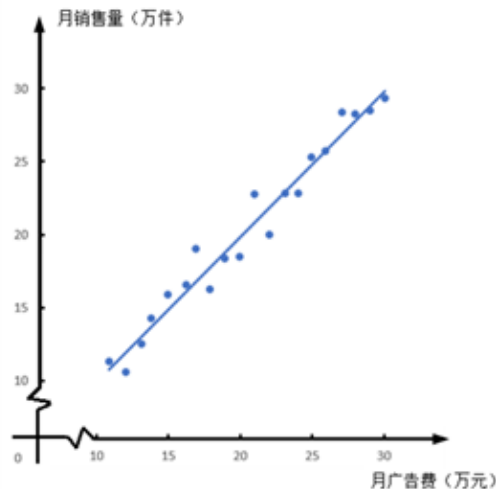
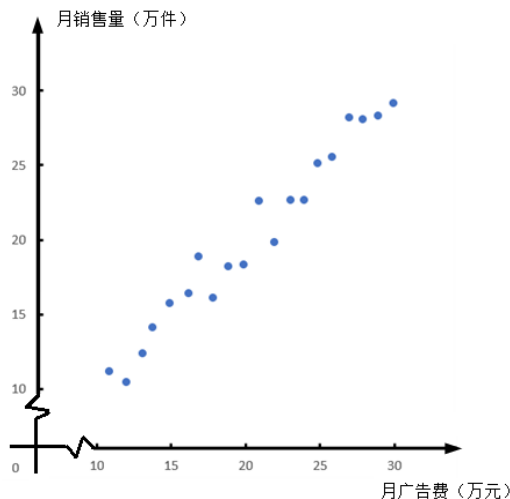
$i$ (月份)	1	2	3	4	5	6	7	8	9	10
$s_i$ (月广告费 万元)	10.95	12.14	13.22	13.87	15.06	16.30	17.01	17.93	19.01	20.01
$t_i$ (月销售量 万件)	11.18	10.43	12.36	14.15	15.73	16.40	18.86	16.13	18.21	18.37
$i$ (月份)	11	12	13	14	15	16	17	18	19	20
$s_i$ (月广告费 万元)	21.04	22.10	23.17	24.07	25.00	25.95	27.10	28.01	29.06	30.05
$t_i$ (月销售量 万件)	22.61	19.83	22.67	22.70	25.16	25.55	28.21	28.12	28.32	29.18



# 线性回归



- 首先将表中的样本数据可视化，如下图所示，不难发现它们基本上成直线排列
- $s_i$ 与 $t_i$ 之间的关系可表示： $f(s) = as + b$
- 令： $x_1 = \psi_1(s) = s$ ； $x_2 = \psi_2(s) = 1$
- 样本 $s_i$ 的特征向量： $(s_i, 1)^T$
- 令 $y = t$ ，则可将表中 $s_i$ 和 $t_i$ 值代入公式 $\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$ 中：
- $\mathbf{w} = (0.968, 0.191)^T$
- 得到线性回归模型： $f(s) = 0.968s + 0.191$



- 多重共线现象

- 多元线性回归模型：其重要假定之一不同样本之间的属性标记值之间**不存在线性关系**。即 $\mathbf{X}^T \mathbf{X}$ 是可逆矩阵
- **多重共线现象**：当矩阵 $\mathbf{X}$ 的行向量之间存在一定的线性相关性时，就会使得矩阵 $\mathbf{X}^T \mathbf{X}$ 不可逆
- **岭回归**：为了解决多重共线现象带来的问题，对线性回归参数的求解方法进行改进

## • 岭回归

- 基本思想：在线性回归模型损失函数上增加一个针对 $\mathbf{w}$ 的范数惩罚函数，通过对目标函数做正则化处理，将参数向量 $\mathbf{w}$ 中所有参数的取值压缩到一个相对较小的范围，即要求 $\mathbf{w}$ 中所有参数的取值不能过大
- 岭回归的损失函数：

$$J(\mathbf{w}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2$$

其中， $\lambda \geq 0$ 称为正则化参数

- 当 $\lambda$ 的取值较大时，惩罚项 $\lambda \mathbf{w}^T \mathbf{w}$ 就会对损失函数的最小化产生一定的干扰，优化算法就会对回归模型参数 $\mathbf{w}$ 赋予较小的取值以消除这种干扰

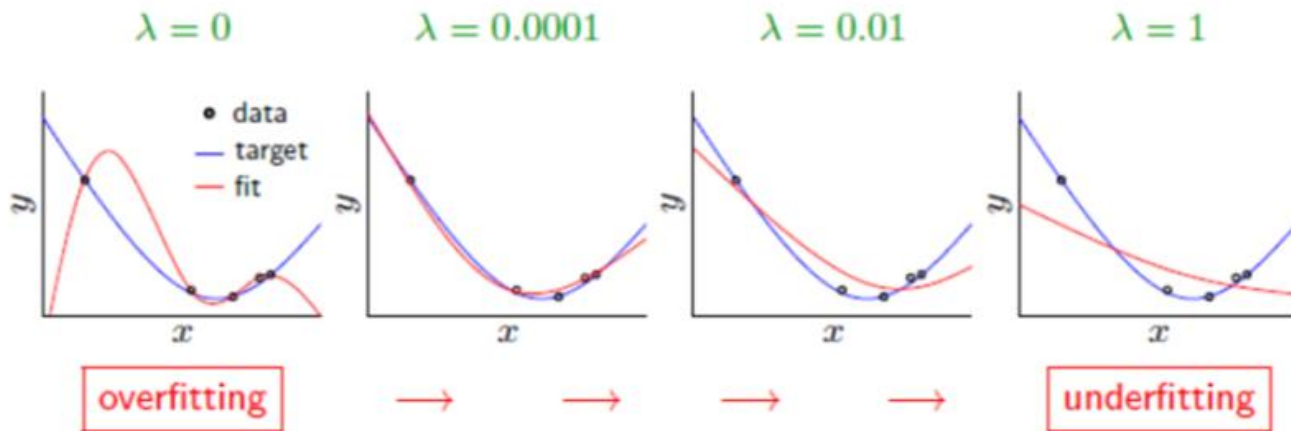
- 岭回归

- 令 $J(\mathbf{w})$ 对参数 $\mathbf{w}$ 的偏导数为0，得：

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$$

其中 $\mathbf{I}$ 为 $m$ 阶单位矩阵，这样即使 $\mathbf{X}^T \mathbf{X}$ 本身不是可逆矩阵，加上 $\lambda \mathbf{I}$ 也可使得 $\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}$ 组成为可逆矩阵

- 正则化参数的影响



# 本节目录



安徽大學  
ANHUI UNIVERSITY



- 线性回归
- **逻辑回归**
- Softmax回归

- 二分类任务

- 预测值与输出标记

$$z = \mathbf{w}^T \mathbf{x} + b \quad y \in \{0, 1\}$$

- 寻找函数将分类标记与线性回归模型输出联系起来
- 最理想的函数——单位阶跃函数

$$y = \begin{cases} 0, & z < 0; \\ 0.5, & z = 0; \\ 1, & z > 0, \end{cases}$$

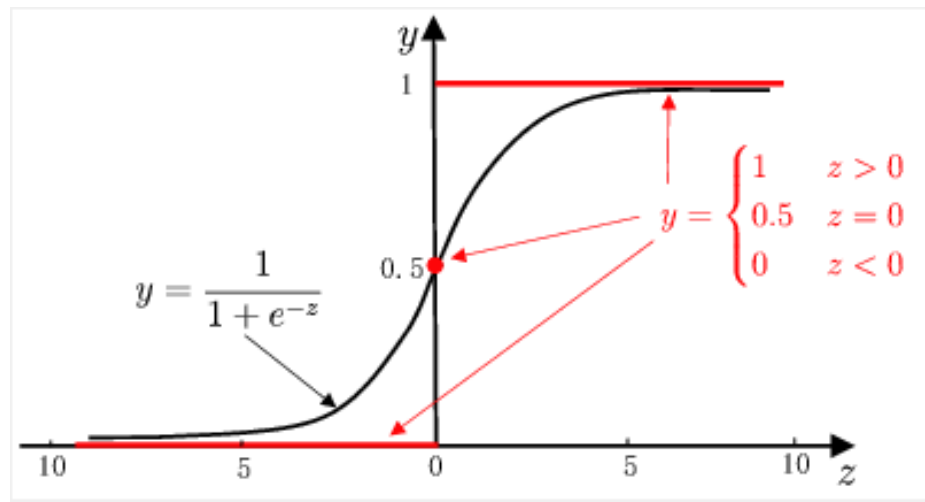
- 预测值大于零就判为正例，小于零就判为反例，预测值为临界值零则可任意判别
- 缺点：不连续

## • 二分类任务

– 替代函数——**逻辑函数** (logistic function)

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

- 单调可微、任意阶可导，可直接应用现有数值优化算法求取最优解
- 直接对分类可能性建模，无需事先假设数据分布
- 可得到“类别”的近似概率预测，对利用概率辅助决策任务有用



- 模型结构

- 线性分类函数可以写成：

$$f(\mathbf{x}; \mathbf{w}, b) = \begin{cases} 1 & \sigma(\mathbf{w}^T \mathbf{x} + b) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

- 可以使用预测值作为分类的可能性，即概率

- 样本 $\mathbf{x}$ 属于正例( $y=1$ )的概率

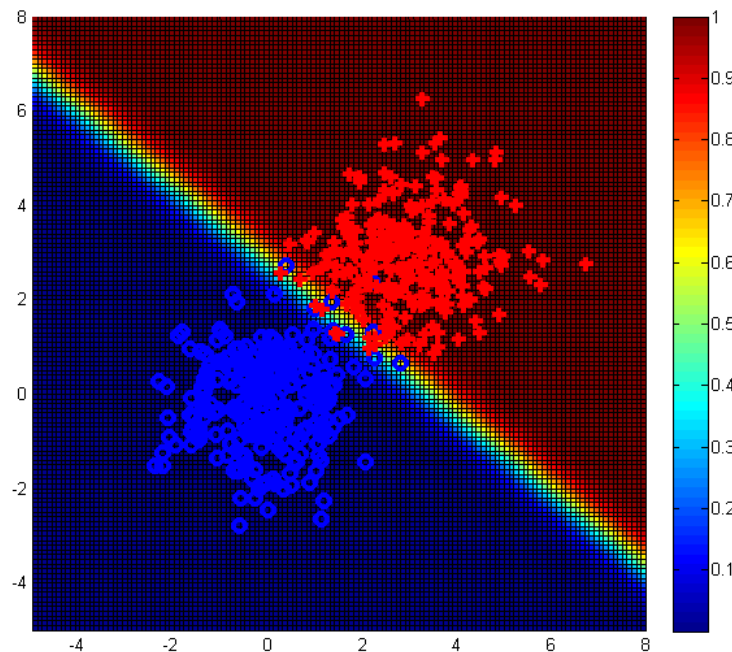
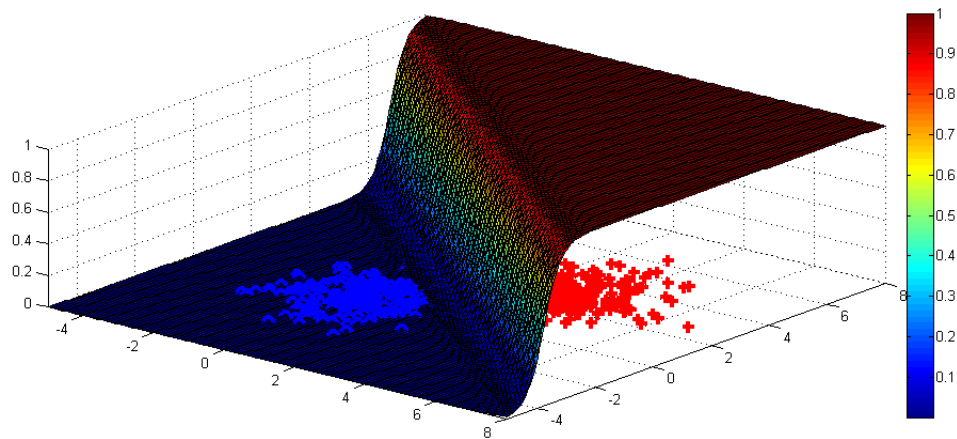
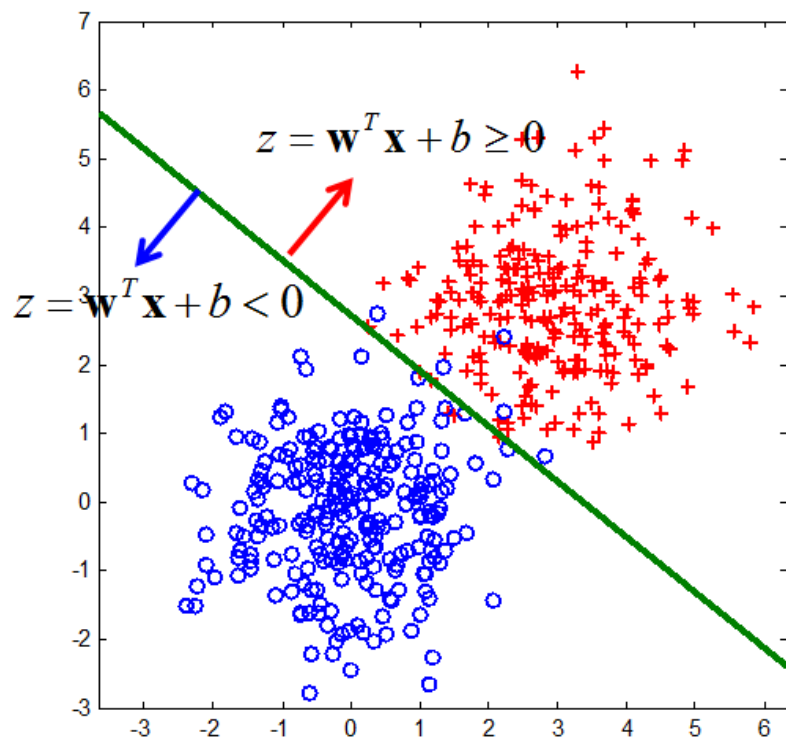
$$P(y=1|\mathbf{x}) = g(\mathbf{x}; \mathbf{w}, b) = \sigma(\mathbf{w}^T \mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}$$

- 样本 $\mathbf{x}$ 属于反例( $y=0$ )的概率

$$P(y=0|\mathbf{x}) = 1 - g(\mathbf{x}; \mathbf{w}, b) = 1 - \sigma(\mathbf{w}^T \mathbf{x} + b) = \frac{e^{-(\mathbf{w}^T \mathbf{x} + b)}}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}}$$



- 模型结构



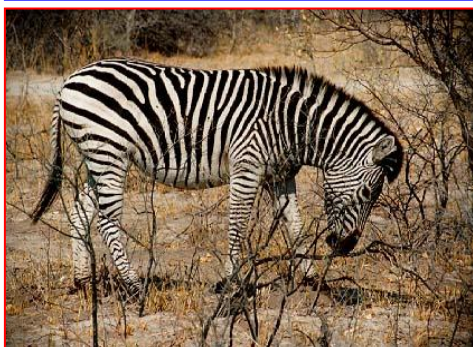
- 模型结构



$$x^{(1)} \rightarrow \begin{bmatrix} 0.1 \\ 0.2 \\ 0.8 \\ \dots \end{bmatrix} \rightarrow P(y^{(1)} \equiv 0 | x^{(1)}) = 0.4$$



$$x^{(2)} \rightarrow \begin{bmatrix} 0.7 \\ 0.5 \\ 0.1 \\ \dots \end{bmatrix} \rightarrow P(y^{(2)} \equiv 0 | x^{(2)}) = 0.2$$



$$x^{(3)} \rightarrow \begin{bmatrix} 0.5 \\ 0.2 \\ 0.9 \\ \dots \end{bmatrix} \rightarrow P(y^{(3)} \equiv 1 | x^{(3)}) = 0.9$$

- 损失函数

- 对数损失（交叉熵损失）

$$P(y | \mathbf{x}) = \begin{cases} g(\mathbf{x}) & y = 1 \\ 1 - g(\mathbf{x}) & y = 0 \end{cases} \Rightarrow$$

$$P(y | \mathbf{x}) = [g(\mathbf{x})]^y [1 - g(\mathbf{x})]^{1-y}$$

$$l(y, h(\mathbf{x})) = -\log P(y | \mathbf{x}) = -y \log g(\mathbf{x}) - (1 - y) \log (1 - g(\mathbf{x}))$$

- 损失函数
  - 经验风险

$$J(\mathbf{w}, b) = R_{emp}(\mathbf{w}, b) = -\frac{1}{m} \sum_{i=1}^m \left\{ y^{(i)} \cdot \log g(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \cdot \log (1 - g(\mathbf{x}^{(i)})) \right\}$$

- 学习算法

- 梯度下降法

$$J(\mathbf{w}, b) = -\frac{1}{m} \sum_{i=1}^m \left\{ y^{(i)} \cdot \log g(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \cdot \log (1 - g(\mathbf{x}^{(i)})) \right\}$$

$$g(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + b) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x} - b)}$$

$$\frac{\partial g}{\partial \mathbf{w}} = g(1 - g) \mathbf{x} \quad \frac{\partial g}{\partial b} = g(1 - g)$$

$$\frac{\partial \log g}{\partial \mathbf{w}} = (1 - g) \mathbf{x} \quad \frac{\partial \log g}{\partial b} = 1 - g$$

$$\frac{\partial \log(1 - g)}{\partial \mathbf{w}} = -g \mathbf{x} \quad \frac{\partial \log(1 - g)}{\partial b} = -g$$

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{1}{m} \sum_{i=1}^m \left( g(\mathbf{x}^{(i)}) - y^{(i)} \right) \mathbf{x}^{(i)}$$

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_{i=1}^m \left( g(\mathbf{x}^{(i)}) - y^{(i)} \right)$$

- 学习算法

- 梯度下降法

用梯度下降法求解Logistic Regression

1. 初始化  $\mathbf{w}, b$
2. 按下式更新  $\mathbf{w}, b$ :

$$\mathbf{w} := \mathbf{w} - \eta \sum_{i=1}^m \left( g(\mathbf{x}^{(i)}) - y^{(i)} \right) \mathbf{x}^{(i)}$$

$$b := b - \eta \sum_{i=1}^m \left( g(\mathbf{x}^{(i)}) - y^{(i)} \right)$$

Batch GD

3. 检查是否收敛，如果不收敛转2

- 学习算法

- 梯度下降法

用梯度下降法求解Logistic Regression

1. 初始化  $\mathbf{w}, b$
2. 按下式更新  $\mathbf{w}, b$ :

*for*  $i = 1:m$

$$\mathbf{w} := \mathbf{w}^{old} - \eta \left( g \left( \mathbf{x}^{(i)}; \mathbf{w}^{old}, b^{old} \right) - y^{(i)} \right) \mathbf{x}^{(i)}$$

$$b := b^{old} - \eta \left( g \left( \mathbf{x}^{(i)}; \mathbf{w}^{old}, b^{old} \right) - y^{(i)} \right)$$

*end*

Stochastic GD

3. 检查是否收敛，如果不收敛转2

- 学习算法

- 梯度下降法

用梯度下降法求解Logistic Regression

1. 初始化  $\mathbf{w}, b$
2. 按下式更新  $\mathbf{w}, b$ :

$$\mathbf{w} := \mathbf{w} - \eta \sum_{i=1}^{batch} \left( g(\mathbf{x}^{(i)}) - y^{(i)} \right) \mathbf{x}^{(i)}$$

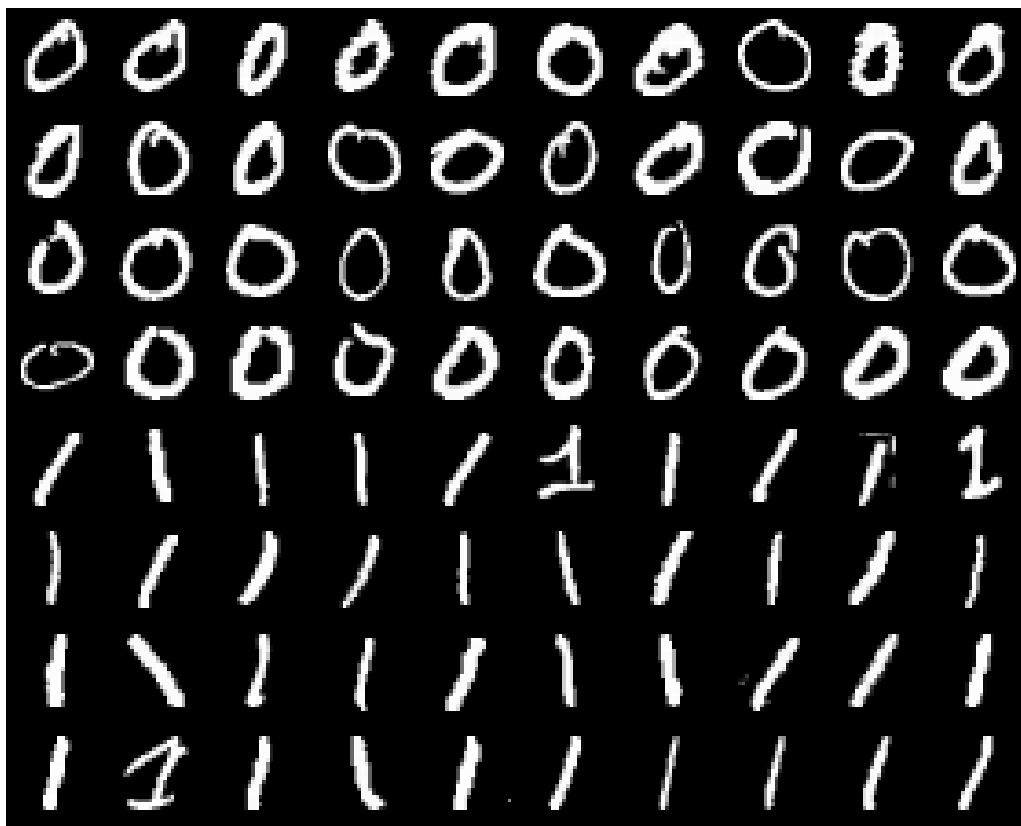
$$b := b - \eta \sum_{i=1}^{batch} \left( g(\mathbf{x}^{(i)}) - y^{(i)} \right)$$

Mini-batch GD

3. 检查是否收敛，如果不收敛转2



- **例题：**使用逻辑回归模型实现手写体图像数字0和1的识别（图像大小均是 $28 \times 28$ ），并对学习结果进行可视化



- 输入一幅图像，产生0和1的识别结果，这是图像二分类问题
- 使用简单的像素值组成的向量作为图像的特征 $\mathbf{x}$ ，由于图像大小是 $28*28$ ，因此特征的维度是784
- 一共有80个训练样本，包括40个正样本( $y=1$ )和40个负样本( $y=0$ )，则我们可以通过逻辑回归算法实现该二分类问题

$$P(y=1|\mathbf{x}) = \frac{1}{1 + \exp(-(\mathbf{w}^T \mathbf{x} + b))}$$

$$P(y=0|\mathbf{x}) = \frac{\exp(-(\mathbf{w}^T \mathbf{x} + b))}{1 + \exp(-(\mathbf{w}^T \mathbf{x} + b))}$$

$$P(y|\mathbf{x}) = [P(y=1|\mathbf{x})]^y [P(y=0|\mathbf{x})]^{1-y}$$

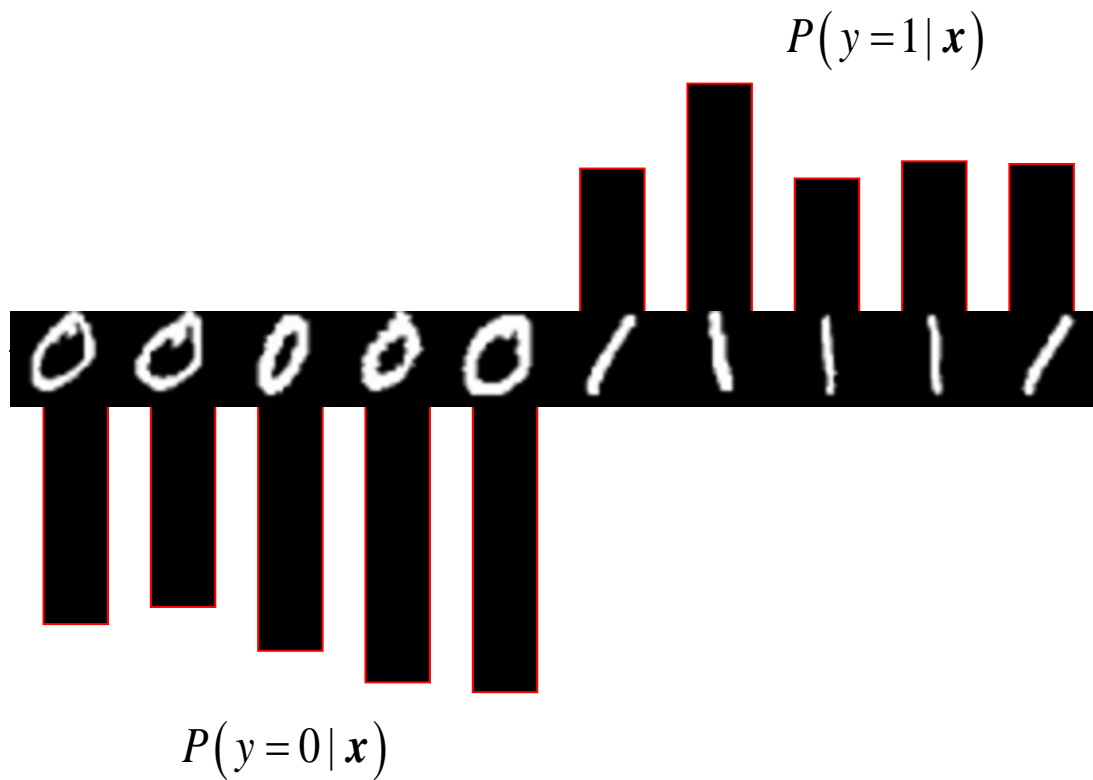
$$J(\mathbf{w}, b; \mathbf{x}, y) = -\frac{1}{m} \sum_{i=1}^m \log P(y_i | \mathbf{x}_i) = \frac{1}{m} \sum_{i=1}^m \log(1 + \exp(-(\mathbf{w}^T \mathbf{x}_i + b))) + (1 - y_i)(\mathbf{w}^T \mathbf{x}_i + b)$$

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{1}{m} \sum_{i=1}^m (g(\mathbf{x}_i) - y_i) \mathbf{x}_i$$

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_{i=1}^m (g(\mathbf{x}_i) - y_i)$$

- 使用梯度下降法求解模型参数

- 识别结果示例



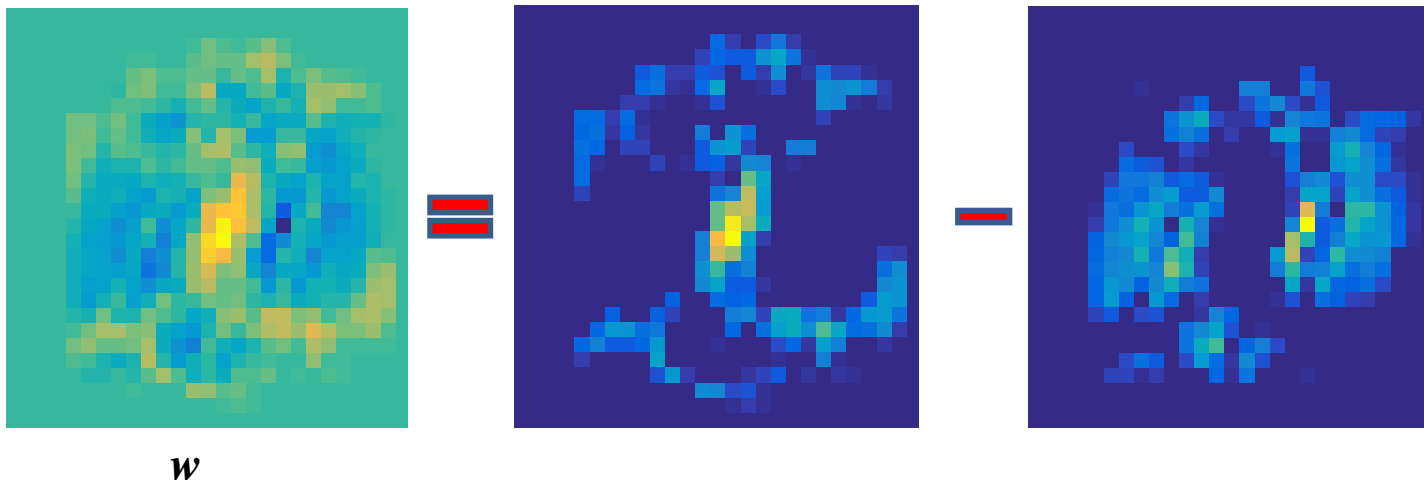
- 结果可视化
  - 如何图形化地展示逻辑回归模型学习的结果？
    - 该模型对何种输入信号响应最强烈

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

$$\mathbf{x} = \arg \max_{\mathbf{x}} \sigma(\mathbf{w}^T \mathbf{x} + b) \quad s.t. \quad \|\mathbf{x}\| = 1$$

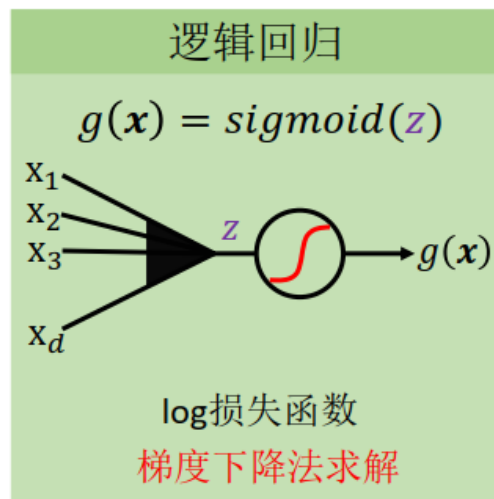
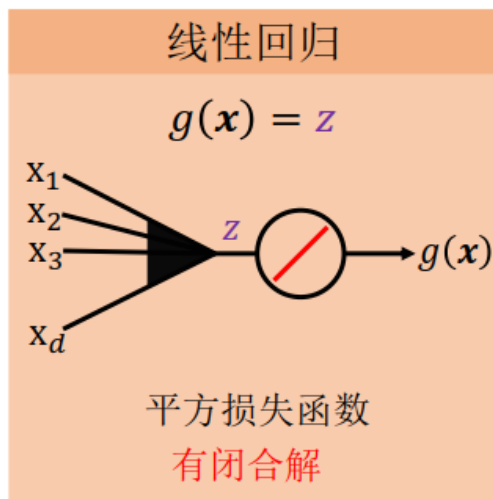
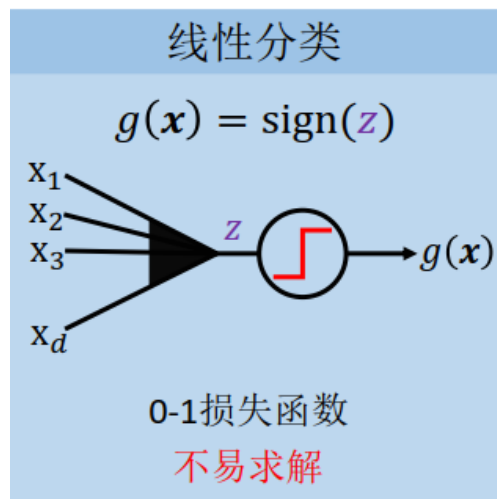
$$\Rightarrow \mathbf{x} = \mathbf{w} / \|\mathbf{w}\|$$

- 可视化结果



- 线性模型总结

## 三种线性模型



$z$ 是关于特征 $\mathbf{x}$ 的线性函数

# 本节目录



安徽大學  
ANHUI UNIVERSITY



- 线性回归
- 逻辑回归
- **Softmax回归**

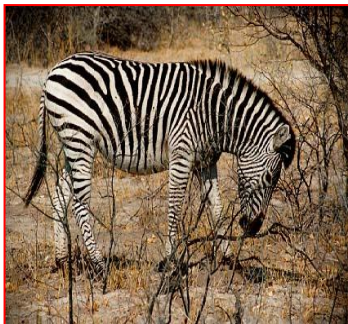
- 逻辑回归的多类拓展



$$\begin{matrix} \longrightarrow & x^{(1)} & \longrightarrow \end{matrix} \begin{bmatrix} 0.1 \\ 0.2 \\ 0.8 \\ \dots \end{bmatrix} \begin{matrix} \longrightarrow & y^{(1)} = 1 \end{matrix}$$



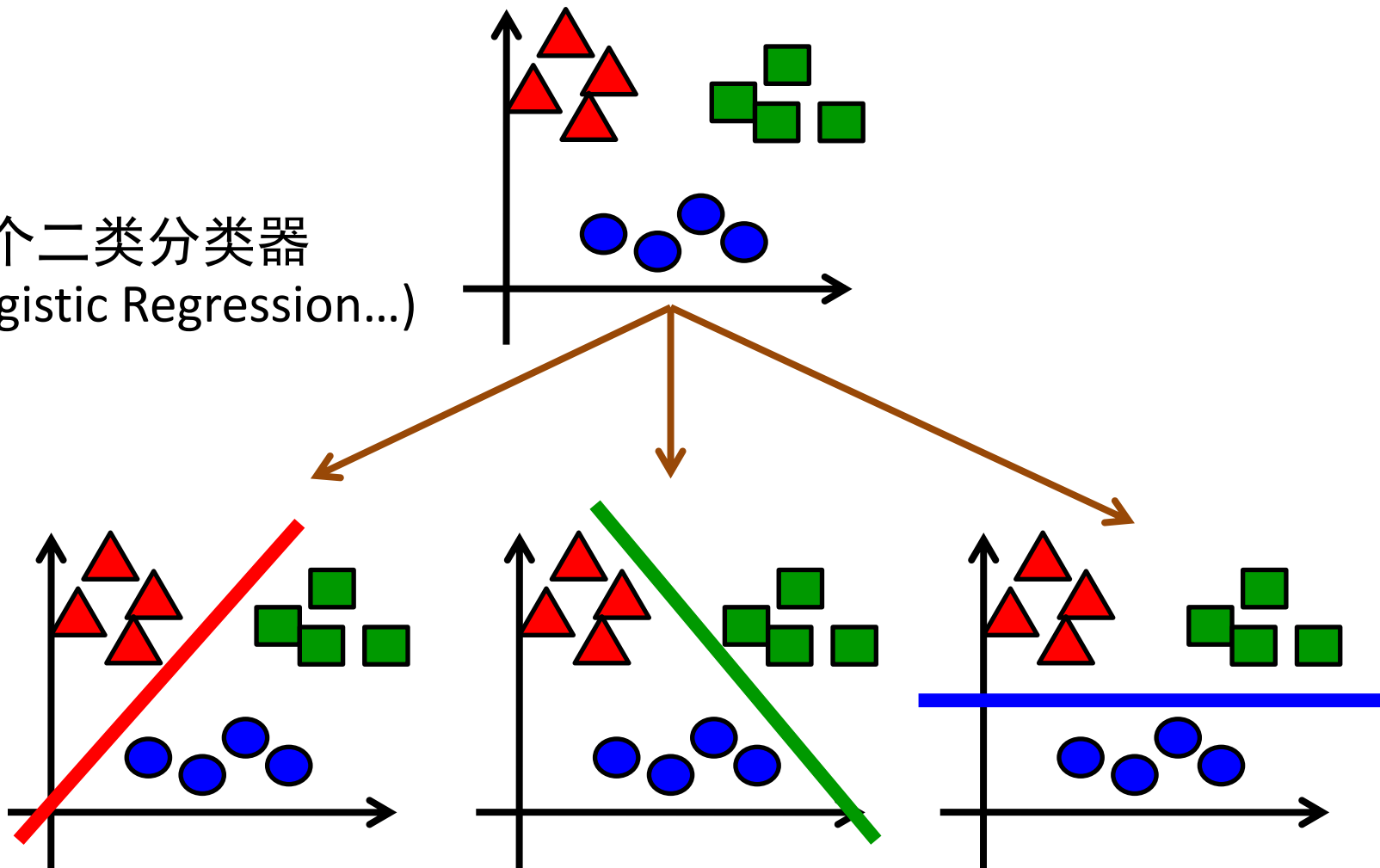
$$\begin{matrix} \longrightarrow & x^{(2)} & \longrightarrow \end{matrix} \begin{bmatrix} 0.7 \\ 0.5 \\ 0.1 \\ \dots \end{bmatrix} \begin{matrix} \longrightarrow & y^{(2)} = 3 \end{matrix}$$



$$\begin{matrix} \longrightarrow & x^{(3)} & \longrightarrow \end{matrix} \begin{bmatrix} 0.5 \\ 0.2 \\ 0.9 \\ \dots \end{bmatrix} \begin{matrix} \longrightarrow & y^{(3)} = 2 \end{matrix}$$

- 逻辑回归的多类拓展

多个二类分类器  
(Logistic Regression...)





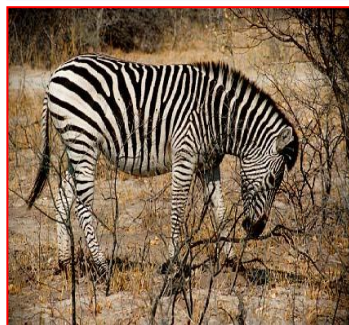
- 逻辑回归的多类拓展



$$\begin{matrix} \longrightarrow & x^{(1)} & \longrightarrow & P(y^{(1)} = 1 | \mathbf{x}^{(1)}) = 0.7 \end{matrix}$$
$$\begin{bmatrix} 0.1 \\ 0.2 \\ 0.8 \\ \dots \end{bmatrix}$$



$$\begin{matrix} \longrightarrow & x^{(2)} & \longrightarrow & P(y^{(2)} = 3 | \mathbf{x}^{(2)}) = 0.9 \end{matrix}$$
$$\begin{bmatrix} 0.7 \\ 0.5 \\ 0.1 \\ \dots \end{bmatrix}$$



$$\begin{matrix} \longrightarrow & x^{(3)} & \longrightarrow & P(y^{(3)} = 2 | \mathbf{x}^{(3)}) = 0.9 \end{matrix}$$
$$\begin{bmatrix} 0.5 \\ 0.2 \\ 0.9 \\ \dots \end{bmatrix}$$

- 逻辑回归的多类拓展
  - 逻辑回归模型

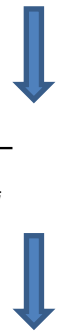
$$\left\{ \begin{array}{l} P(y=1|\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + b) = \frac{1}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}} \\ P(y=0|\mathbf{x}) = 1 - \sigma(\mathbf{w}^T \mathbf{x} + b) = \frac{e^{-(\mathbf{w}^T \mathbf{x} + b)}}{1 + e^{-(\mathbf{w}^T \mathbf{x} + b)}} \end{array} \right.$$

$$\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = e^{\mathbf{w}^T \mathbf{x} + b} \Rightarrow \text{logit} = \log \frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})} = \mathbf{w}^T \mathbf{x} + b$$

- 逻辑回归的多类拓展

- 扩展到 $k$ 个类别

$$\log \frac{P(y=j|\mathbf{x})}{P(y=1|\mathbf{x})} = \mathbf{w}_j^T \mathbf{x} + b_j \Rightarrow P(y=j|\mathbf{x}) = e^{\mathbf{w}_j^T \mathbf{x} + b_j} P(y=1|\mathbf{x}), \quad j=1..k$$


$$P(y=1|\mathbf{x}) = \frac{1}{1 + \sum_{j=2}^k e^{\mathbf{w}_j^T \mathbf{x} + b_j}} \quad P(y=j|\mathbf{x}) = \frac{e^{\mathbf{w}_j^T \mathbf{x} + b_j}}{1 + \sum_{c=2}^k e^{\mathbf{w}_c^T \mathbf{x} + b_c}}$$
$$P(y=j|\mathbf{x}) = \frac{e^{\mathbf{w}_j^T \mathbf{x} + b_j}}{\sum_{c=1}^k e^{\mathbf{w}_c^T \mathbf{x} + b_c}}$$

- 样本 $x$ 属于第 $k$ 类( $y=k$ )的概率:

$$P(y=k|\mathbf{x}) = \frac{\exp(\boldsymbol{\theta}_k^T \mathbf{x})}{\sum_{j=1}^C \exp(\boldsymbol{\theta}_j^T \mathbf{x})} \in (0, 1), \quad \boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_c\}$$

- 损失函数与经验风险

- 对数损失

$$\begin{aligned}l(y, h(\mathbf{x})) &= -\log P(y | \mathbf{x}; \boldsymbol{\theta}) = -\log \left\{ \prod_{c=1}^k [P(y = c | \mathbf{x}; \boldsymbol{\theta})]^{I(y=c)} \right\} \\&= \log \sum_{c=1}^k e^{(\mathbf{w}_c^T \mathbf{x} + b_c)} - \sum_{c=1}^k I(y = c) (\mathbf{w}_c^T \mathbf{x} + b_c)\end{aligned}$$

- 经验风险

$$J(\boldsymbol{\theta}) = R_{emp}(\boldsymbol{\theta}) = \frac{1}{m} \left\{ \sum_{i=1}^m l(y_i, h(\mathbf{x}_i)) \right\}$$

- 损失函数与经验风险

- 参数冗余

$$\theta = \{\theta_1, \theta_2, \dots, \theta_c\} \Rightarrow \theta' = \{\theta_1 - \phi, \theta_2 - \phi, \dots, \theta_c - \phi\}$$

$$g(\mathbf{x}; \theta) = \frac{1}{\sum_{j=1}^c \exp(\theta_j^T \mathbf{x})} \begin{bmatrix} \exp(\theta_1^T \mathbf{x}) \\ \exp(\theta_2^T \mathbf{x}) \\ \vdots \\ \exp(\theta_c^T \mathbf{x}) \end{bmatrix} =$$
$$\frac{1}{\sum_{j=1}^c \exp((\theta_j - \phi)^T \mathbf{x})} \begin{bmatrix} \exp((\theta_1 - \phi)^T \mathbf{x}) \\ \exp((\theta_2 - \phi)^T \mathbf{x}) \\ \vdots \\ \exp((\theta_c - \phi)^T \mathbf{x}) \end{bmatrix} = g(\mathbf{x}; \theta')$$

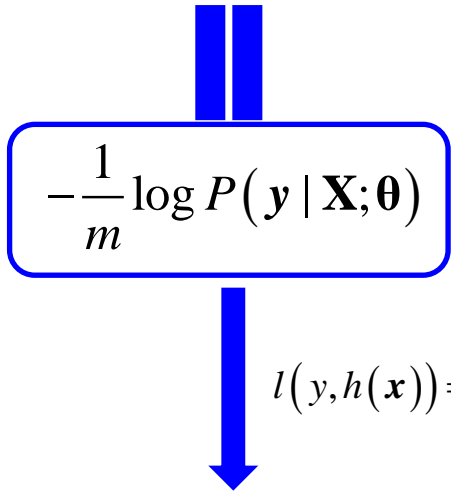
- 结构风险：加入控制模型复杂度的正则化项

$$J(\theta) = -\frac{1}{m} \left\{ \sum_{i=1}^m l(y_i, h(\mathbf{x}_i)) \right\} + \underbrace{\frac{\lambda}{2} \sum_{k=1}^c \theta_k^T \theta_k}_{\text{权值衰减项 (Weight Decay)}}$$

- 学习算法

- 梯度下降法

$$J(\boldsymbol{\theta}) = \frac{1}{m} \left\{ \sum_{i=1}^m l(y_i, h(\mathbf{x}_i)) \right\} + \frac{\lambda}{2} \sum_{k=1}^C \boldsymbol{\theta}_k^T \boldsymbol{\theta}_k$$


$$-\frac{1}{m} \log P(y | \mathbf{X}; \boldsymbol{\theta})$$

$$l(y, h(\mathbf{x})) = \log \sum_{c=1}^k e^{(\mathbf{w}_c^T \mathbf{x} + b_c)} - \sum_{c=1}^k I(y = c)(\mathbf{w}_c^T \mathbf{x} + b_c)$$

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_j} = \frac{1}{m} \sum_{i=1}^m \left[ \left( \frac{e^{\boldsymbol{\theta}_j^T \mathbf{x}^{(i)}}}{\sum_{c=1}^k e^{\boldsymbol{\theta}_c^T \mathbf{x}^{(i)}}} - I(y^{(i)} = j) \right) \mathbf{x}^{(i)} \right] + \lambda \boldsymbol{\theta}_j$$

- **例题：**使用Softmax回归模型实现手写体图像数字识别（图像大小均是 $28*28$ ）



- 输入为一幅图像，产生0-9的识别结果，这是图像多分类问题
- 使用简单的像素值组成的向量作为图像的特征 $\mathbf{x}$ ，由于图像大小是28\*28，因此特征的维度是784。输出标记的维度是10

$$P(y = k | \mathbf{x}) = \frac{\exp(\boldsymbol{\theta}_k^T \mathbf{x})}{\sum_{i=1}^{10} \exp(\boldsymbol{\theta}_i^T \mathbf{x})} \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{10} \end{pmatrix}; y_k = \begin{cases} 1 & x \in k \\ 0 & x \notin k \end{cases}$$

$$P(y | \mathbf{x}) = \prod_{j=1}^{10} [P(y = j | \mathbf{x})]^{y=j}$$

$$J(\boldsymbol{\theta}; \mathbf{x}, y) = -\log P(y | \mathbf{x}) = -\sum_{j=1}^{10} \left\{ y_j \log [a_j^{(3)}(x)] \right\}$$

$$J(\boldsymbol{\theta}) = \frac{1}{100} \sum_{i=1}^{100} \sum_{j=1}^{10} \left[ P(y^{(i)} = j | \mathbf{x}^{(i)}; \boldsymbol{\theta}_j) - I(y^{(i)} = j) \right] + \frac{\lambda}{200} \left\{ \sum_{i=1}^{100} \sum_{j=1}^{10} \|\boldsymbol{\theta}_j^{(i)}\|^2 \right\}$$

- 使用梯度下降法求解模型参数



# 本节目录



安徽大學  
ANHUI UNIVERSITY



- **线性回归**
  - 最小二乘法
  - 岭回归
- **逻辑回归**
  - 逻辑函数
  - 软性二分类模型
  - 对数损失
  - 学习算法
- **Softmax回归**
  - 逻辑回归的多类拓展
  - 对数损失
  - 结构风险
  - 学习算法

# 思考题



安徽大学  
ANHUI UNIVERSITY



- 对于多分类问题，Softmax回归模型一定比逻辑回归模型（分解成多个二分类任务）更合适吗
- SVM与逻辑回归模型有哪些区别？最本质区别是什么？

# 练习题



安徽大學  
ANHUI UNIVERSITY



- 现有10组正样本和10组负样本组成的手写体图像数字识别训练集，以及10组正样本和10组负样本组成的测试集。试用逻辑回归算法实现手写体数字图像识别，计算测试集精度，给出训练过程中的收敛曲线，并对模型进行可视化分析（设定参数：迭代次数：3，学习率：0.1）