

# Machine Learning Formulas

MingShun Wu

September 17, 2016

# Contents

1	线性回归(Linear Regression)	3
1.1	当训练集X只有1项时	3
1.2	当训练集X有m项时	3
1.2.1	Cost Function	4
1.2.2	梯度下降	4
1.3	Feature Normalization	5
1.4	公式法求解 (Normal Equation)	5
2	逻辑回归(Logistic Regression)	6
2.1	当只有2个类别时, 使用1个分类器	6
2.1.1	预测函数	6
2.1.2	Cost Function	6
2.1.3	梯度下降	6
2.2	当只有k个类别时, 使用k个分类器	6
3	Regularization	7
3.1	线性回归	7
3.1.1	数值计算方式	7
3.1.2	矩阵计算方式	7
3.2	逻辑回归	7
3.2.1	数值计算方式	7
3.2.2	矩阵计算方式	7
3.3	注意	7
4	神经网络 - 前向算法	8
4.1	神经网络示意图 - 前向算法	8
4.2	神经网络 - 前向算法	10
4.2.1	$X$ 、 $\theta$ 、 $\Theta$ 、 $z$ 、 $a$	10
4.2.2	$y$	12
5	神经网络 - 后向算法	14
5.1	神经网络示意图 - 后向算法	14
5.2	神经网络 - 后向算法	15
5.2.1	输出层结果: $a^L$	15
5.2.2	格式化后的Y	15
5.2.3	$\delta^L$	15
5.2.4	$\delta^{L-1}$	15
5.2.5	$\delta^l (2 \leq l \leq L - 2)$	15
5.2.6	$\Delta^l$ (用迭代的方式计算)	16
5.2.7	$D_{ij}^{(l)}$	16
5.2.8	$\frac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}}$	16
5.2.9	$\delta^l$ 与 $\Delta^l$ 的区别与联系	16

# 1 线性回归(Linear Regression)

## 1.1 当训练集X只有1项时

$$X = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}_{(n+1)*1} \quad (1)$$

$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{pmatrix}_{(n+1)*1} \quad (2)$$

$$y = y \quad (3)$$

$$\begin{aligned} h_{\theta}(x) &= \theta^T X = X^T \theta \\ &= \begin{pmatrix} 1 & x_1 & x_2 & \dots & x_n \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{pmatrix} \\ &= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \end{aligned} \quad (4)$$

## 1.2 当训练集X有m项时

$$\begin{aligned} X &= \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ \vdots \\ x^{(m)} \end{pmatrix} \\ &= \begin{pmatrix} x_0^{(1)} & x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \dots & x_n^{(1)} \\ x_0^{(2)} & x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \dots & x_n^{(2)} \\ x_0^{(3)} & x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & \dots & x_n^{(3)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_0^{(m)} & x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \\ &= \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \dots & x_n^{(2)} \\ 1 & x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & \dots & x_n^{(3)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \dots & x_n^{(m)} \end{pmatrix}_{m*(n+1)} \end{aligned} \quad (5)$$

$$\theta = \begin{pmatrix} \theta^{(0)} \\ \theta^{(1)} \\ \theta^{(2)} \\ \theta^{(3)} \\ \vdots \\ \theta^{(n)} \end{pmatrix}_{(n+1)*1} \quad (6)$$

$$h_{\theta}(x) = X * \theta$$

$$= \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \dots & x_n^{(2)} \\ 1 & x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & \dots & x_n^{(3)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} \theta^{(0)} \\ \theta^{(1)} \\ \theta^{(2)} \\ \theta^{(3)} \\ \vdots \\ \theta^{(n)} \end{pmatrix} \quad (7)$$

$$y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ \vdots \\ y^{(m)} \end{pmatrix}_{m*1} \quad (8)$$

### 1.2.1 Cost Function

1. 数值形式:

$$J(\theta) = \frac{1}{2m} [h_{\theta}(x^{(i)}) - y^{(i)}]^2 \quad (9)$$

2. 矩阵形式:

$$J(\theta) = \frac{1}{2m} [h_{\theta}(x) - y]^T [h_{\theta}(x) - y] \quad (10)$$

### 1.2.2 梯度下降

1. 数值形式

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} [h_{\theta}(x^{(i)}) - y^{(i)}] x_j^{(i)} \quad (11)$$

迭代方式:

$$\theta_j := \theta_j - \alpha \frac{1}{m} [h_{\theta}(x^{(i)}) - y^{(i)}] x_j^{(i)} \quad (12)$$

2. 矩阵形式

$$\nabla J(\theta) = \frac{1}{2m} X^T [h_{\theta}(x) - y] \quad (13)$$

迭代方式:

$$\theta := \theta - \alpha \frac{1}{m} X^T [h_{\theta}(x) - y] \quad (14)$$

### 1.3 Feature Normalization

$$x_i = \frac{x_i - \mu}{\sigma} \quad (15)$$

或

$$x_i = \frac{x_i - \mu}{max - min} \quad (16)$$

### 1.4 公式法求解 (Normal Equation)

$$\theta = (X^T X)^{-1} X^T y \quad (17)$$

## 2 逻辑回归(Logistic Regression)

### 2.1 当只有2个类别时，使用1个分类器

#### 2.1.1 预测函数

##### 1. 数值形式

$$h_{\theta}(x) = \frac{1}{1 + e^{\theta^T x}} \quad (18)$$

##### 2. 矩阵形式

$$h_{\theta}(X) = \frac{1}{1 + e^{X\theta}} \quad (19)$$

#### 2.1.2 Cost Function

##### 1. 数值形式

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m [-y^{(i)} \log h_{\theta}(x^{(i)}) - (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))] \quad (20)$$

##### 2. 矩阵形式

$$J(\theta) = \frac{1}{m} [-y^T \log h_{\theta}(x) - (1 - y^T) \log(1 - h_{\theta}(x))] \quad (21)$$

#### 2.1.3 梯度下降

##### 1. 数值形式

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m [h_{\theta}(x^{(i)}) - y^{(i)}] x_j^{(i)} \quad (22)$$

迭代方式:

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m [h_{\theta}(x^{(i)}) - y^{(i)}] x_j^{(i)} \quad (23)$$

##### 2. 矩阵形式

$$\nabla J(\theta) = \frac{1}{m} X^T [h_{\theta}(x) - y] \quad (24)$$

迭代方式:

$$\theta := \theta - \alpha \frac{1}{m} X^T [h_{\theta}(x) - y] \quad (25)$$

### 2.2 当只有k个类别时，使用k个分类器

### 3 Regularization

#### 3.1 线性回归

##### 3.1.1 数值计算方式

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m [h_{\theta}(x^{(i)}) - y^{(i)}]^2 + \lambda \frac{1}{2m} \sum_{j=1}^n \theta_j^2 \quad (26)$$

$$\begin{aligned} \frac{\partial J(\theta)}{\partial \theta_j} &= \frac{1}{2m} \sum_{i=1}^m 2[h_{\theta}(x^{(i)}) - y^{(i)}] \frac{\partial h_{\theta}(x^{(i)})}{\partial \theta_j} + \lambda \frac{1}{2m} 2 \sum_{i=1}^n \theta_j \\ &= \frac{1}{m} \sum_{i=1}^m [h_{\theta}(x^{(i)}) - y^{(i)}] x_j^{(i)} + \frac{\lambda}{m} \sum_{i=1}^n \theta_j \end{aligned} \quad (27)$$

$$\begin{cases} \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_0^{(i)}, & j = 0 \\ \theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} + \alpha \frac{\lambda}{m} \theta_j, & j \neq 0 \end{cases}$$

##### 3.1.2 矩阵计算方式

$$J(\theta) = \frac{1}{2m} [h_{\theta}(x) - y]^T [h_{\theta}(x) - y] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2 \quad (28)$$

$$\nabla J(\theta) = \frac{1}{2m} X^T [h_{\theta}(x) - y] + \frac{\lambda}{m} \sum_{i=1}^n \theta_j \quad (29)$$

$$\text{matlab} \begin{cases} \text{grad} & = 1/m * X' * (h - y);, \\ \text{grad}(2 : \text{end}) & = \text{grad}(2 : \text{end}) + \text{lambda}/m * \text{theta}(2 : \text{end}); \end{cases}$$

#### 3.2 逻辑回归

##### 3.2.1 数值计算方式

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m [-y^{(i)} \log h_{\theta}(x^{(i)}) - (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))] + \lambda \frac{1}{2m} \sum_{j=1}^n \theta_j^2 \quad (30)$$

$$\begin{cases} \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_0^{(i)}, & j = 0 \\ \theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} + \alpha \frac{\lambda}{m} \theta_j, & j \neq 0 \end{cases}$$

##### 3.2.2 矩阵计算方式

$$J(\theta) = \frac{1}{m} [-y^T \log h_{\theta}(x) - (1 - y^T) \log(1 - h_{\theta}(x))] + \lambda \frac{1}{2m} \theta^T \theta \quad (31)$$

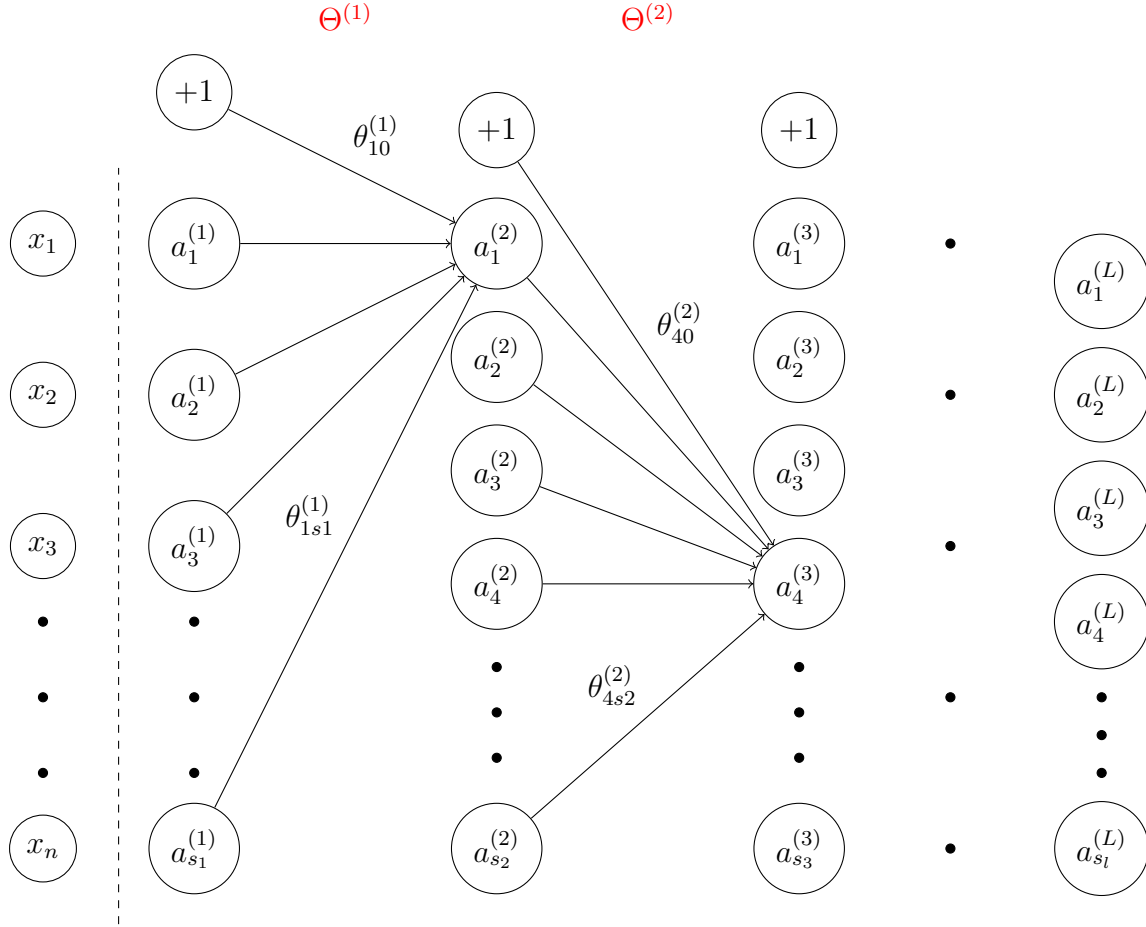
$$\text{matlab} \begin{cases} \text{grad} & = 1/m * X' * (h - y);, \\ \text{grad}(2 : \text{end}) & = \text{grad}(2 : \text{end}) + \text{lambda}/m * \text{theta}(2 : \text{end}); \end{cases}$$

#### 3.3 注意

在实际计算 $\theta$ 中，都是先计算没有Regularization的结果，再对(2:end)计算有Regularization的结果，再将其回到没有Regularization的结果中

## 4 神经网络 - 前向算法

### 4.1 神经网络示意图 - 前向算法



$$a^{(j)} = g(z^{(j-1)}) \Rightarrow (m, s_j) \quad (32)$$

$$\Theta^{(j)} = \begin{pmatrix} \theta_{10}^{(j)} & \theta_{11}^{(j)} & \theta_{12}^{(j)} & \cdots & \theta_{1,s_j}^{(j)} \\ \theta_{20}^{(j)} & \theta_{21}^{(j)} & \theta_{22}^{(j)} & \cdots & \theta_{2,s_j}^{(j)} \\ \theta_{30}^{(j)} & \theta_{31}^{(j)} & \theta_{32}^{(j)} & \cdots & \theta_{3,s_j}^{(j)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \theta_{s_{j+1}0}^{(j)} & \theta_{s_{j+1}1}^{(j)} & \theta_{s_{j+1}2}^{(j)} & \cdots & \theta_{s_{j+1},s_j}^{(j)} \end{pmatrix} \Rightarrow (s_{j+1}, s_j + 1) \quad (33)$$

$$z^{(j+1)} = (1, a^{(j)})(\Theta^{(j)})^T$$

$$= \begin{pmatrix} z_1^{(j+1)(1)} & z_2^{(j+1)(1)} & z_3^{(j+1)(1)} & \cdots & z_{s_{j+1}}^{(j+1)(1)} \\ z_1^{(j+1)(2)} & z_2^{(j+1)(2)} & z_3^{(j+1)(2)} & \cdots & z_{s_{j+1}}^{(j+1)(2)} \\ z_1^{(j+1)(3)} & z_2^{(j+1)(3)} & z_3^{(j+1)(3)} & \cdots & z_{s_{j+1}}^{(j+1)(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_1^{(j+1)(m)} & z_2^{(j+1)(m)} & z_3^{(j+1)(m)} & \cdots & z_{s_{j+1}}^{(j+1)(m)} \end{pmatrix} \quad (34)$$

$$a^{(j+1)} = g(z^{(j+1)}) \Rightarrow (m, s_{j+1}) \quad (35)$$



$$y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ \vdots \\ y^{(m)} \end{pmatrix}_{m \times 1} \tag{36}$$

## 4.2 神经网络 - 前向算法

### 4.2.1 $X$ 、 $\theta$ 、 $\Theta$ 、 $z$ 、 $a$

1.  $X$

$$\begin{aligned}
 X &= \begin{pmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ (x^{(3)})^T \\ \vdots \\ (x^{(m)})^T \end{pmatrix} \\
 &= \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \dots & x_n^{(2)} \\ x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & \dots & x_n^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \\
 &= \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \dots & x_n^{(2)} \\ x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & \dots & x_n^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \Rightarrow (m, n)
 \end{aligned} \tag{37}$$

2.  $a^{(1)}$

$$a^{(1)} = X \Rightarrow (m, n) \tag{38}$$

3.  $\Theta^{(1)}$

$$\Theta^{(1)} = \begin{pmatrix} \theta_{10}^{(1)} & \theta_{11}^{(1)} & \theta_{12}^{(1)} & \dots & \theta_{1,s_1}^{(1)} \\ \theta_{20}^{(1)} & \theta_{21}^{(1)} & \theta_{22}^{(1)} & \dots & \theta_{2,s_1}^{(1)} \\ \theta_{30}^{(1)} & \theta_{31}^{(1)} & \theta_{32}^{(1)} & \dots & \theta_{3,s_1}^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \theta_{s_2 0}^{(1)} & \theta_{s_2 1}^{(1)} & \theta_{s_2 2}^{(1)} & \dots & \theta_{s_2, s_1}^{(1)} \end{pmatrix} \Rightarrow (s_2, s_1 + 1) = (s_2, n + 1) \tag{39}$$

4.  $z^{(2)}$

给 $a^{(1)}$ 的每个数据均添加上 $a_0 = 1$ 后与 $\Theta^{(1)}$ 计算,得到 $z^{(2)\text{注}[1]} = (1, a^{(1)})(\Theta^{(1)})^T$

$$\begin{aligned}
z^{(2)} &= (1, a^{(1)})(\Theta^{(1)})^T \Rightarrow (m, n+1) * (n+1, s_2) \\
&= \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \dots & x_n^{(2)} \\ 1 & x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & \dots & x_n^{(3)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} \theta_{10}^{(1)} & \theta_{11}^{(1)} & \theta_{12}^{(1)} & \dots & \theta_{1,n}^{(1)} \\ \theta_{20}^{(1)} & \theta_{21}^{(1)} & \theta_{22}^{(1)} & \dots & \theta_{2,n}^{(1)} \\ \theta_{30}^{(1)} & \theta_{31}^{(1)} & \theta_{32}^{(1)} & \dots & \theta_{3,n}^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \theta_{s_2,0}^{(j)} & \theta_{s_2,1}^{(j)} & \theta_{s_2,2}^{(j)} & \dots & \theta_{s_2,n}^{(1)} \end{pmatrix}^T \\
&= \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \dots & x_n^{(2)} \\ 1 & x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & \dots & x_n^{(3)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \begin{pmatrix} \theta_{10}^{(1)} & \theta_{20}^{(1)} & \theta_{30}^{(1)} & \dots & \theta_{s_2,0}^{(1)} \\ \theta_{11}^{(1)} & \theta_{21}^{(1)} & \theta_{31}^{(1)} & \dots & \theta_{s_2,1}^{(1)} \\ \theta_{12}^{(1)} & \theta_{22}^{(1)} & \theta_{32}^{(1)} & \dots & \theta_{s_2,2}^{(1)} \\ \theta_{13}^{(1)} & \theta_{23}^{(1)} & \theta_{33}^{(1)} & \dots & \theta_{s_2,3}^{(1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \theta_{1,n}^{(1)} & \theta_{2,n}^{(1)} & \theta_{3,n}^{(1)} & \dots & \theta_{s_2,n}^{(1)} \end{pmatrix} \quad (40) \\
&= \begin{pmatrix} z^{(2)(1)} \\ z^{(2)(2)} \\ z^{(2)(3)} \\ \vdots \\ z^{(2)(m)} \end{pmatrix} \\
&= \begin{pmatrix} z_1^{(2)(1)} & z_2^{(2)(1)} & z_3^{(2)(1)} & \dots & z_{s_2}^{(2)(1)} \\ z_1^{(2)(2)} & z_2^{(2)(2)} & z_3^{(2)(2)} & \dots & z_{s_2}^{(2)(2)} \\ z_1^{(2)(3)} & z_2^{(2)(3)} & z_3^{(2)(3)} & \dots & z_{s_2}^{(2)(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_1^{(2)(m)} & z_2^{(2)(m)} & z_3^{(2)(m)} & \dots & z_{s_2}^{(2)(m)} \end{pmatrix} \\
&\Rightarrow (m, n+1) * (n+1, s_2) = (m, s_2)
\end{aligned}$$

注[2]

5.  $a^{(2)}$

$$a^{(2)} = g(z^{(2)}) \Rightarrow (m, s_2) \quad (41)$$

6. 后续同理

$$\begin{aligned}
\Theta^{(2)} &= \begin{pmatrix} \theta_{10}^{(2)} & \theta_{11}^{(2)} & \theta_{12}^{(2)} & \dots & \theta_{1,s_2}^{(2)} \\ \theta_{20}^{(2)} & \theta_{21}^{(2)} & \theta_{22}^{(2)} & \dots & \theta_{2,s_2}^{(2)} \\ \theta_{30}^{(2)} & \theta_{31}^{(2)} & \theta_{32}^{(2)} & \dots & \theta_{3,s_2}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \theta_{s_3,0}^{(2)} & \theta_{s_3,1}^{(2)} & \theta_{s_3,2}^{(2)} & \dots & \theta_{s_3,s_2}^{(2)} \end{pmatrix} \Rightarrow (s_3, s_2+1) \\
z^{(3)} &= (1, a^{(2)})(\Theta^{(2)})^T \Rightarrow (m, s_2+1) * (s_2+1, s_3) = (m, s_3) \\
a^{(3)} &= g(z^{(3)}) \Rightarrow (m, s_3) \\
&\vdots
\end{aligned} \quad (42)$$

注[1]从 $a^{(1)}$ 得到 $a^{(2)}$ 需要经过sigmoid()函数, 后续的从 $a^{(j)}$ 得到 $a^{(j+1)}$ 均需要经过sigmoid()函数

注[2]上式 $z_{s_2}^{(2)(m)}$ 中, (2)表示第2层神经网络, (m)表示第m个训练集,  $s_2$ 表示第2层神经网络的最后一个单元

## 7. 一般式

$$\begin{aligned}
a^{(j)} &= g(z^{(j-1)}) \Rightarrow (m, s_j) \\
\Theta^{(j)} &= \begin{pmatrix} \theta_{10}^{(j)} & \theta_{11}^{(j)} & \theta_{12}^{(j)} & \cdots & \theta_{1,s_j}^{(j)} \\ \theta_{20}^{(j)} & \theta_{21}^{(j)} & \theta_{22}^{(j)} & \cdots & \theta_{2,s_j}^{(j)} \\ \theta_{30}^{(j)} & \theta_{31}^{(j)} & \theta_{32}^{(j)} & \cdots & \theta_{3,s_j}^{(j)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \theta_{s_{j+1}0}^{(j)} & \theta_{s_{j+1}1}^{(j)} & \theta_{s_{j+1}2}^{(j)} & \cdots & \theta_{s_{j+1},s_j}^{(j)} \end{pmatrix} \Rightarrow (s_{j+1}, s_j + 1) \\
z^{(j+1)} &= (1, a^{(j)})(\Theta^{(j)})^T \\
&= \begin{pmatrix} 1 & a_1^{(j)(1)} & a_2^{(j)(1)} & a_3^{(j)(1)} & \cdots & a_{s_j}^{(j)(1)} \\ 1 & a_1^{(j)(2)} & a_2^{(j)(2)} & a_3^{(j)(2)} & \cdots & a_{s_j}^{(j)(2)} \\ 1 & a_1^{(j)(3)} & a_2^{(j)(3)} & a_3^{(j)(3)} & \cdots & a_{s_j}^{(j)(3)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_1^{(j)(m)} & a_2^{(j)(m)} & a_3^{(j)(m)} & \cdots & a_{s_j}^{(j)(m)} \end{pmatrix} \begin{pmatrix} \theta_{10}^{(j)} & \theta_{20}^{(j)} & \theta_{30}^{(j)} & \cdots & \theta_{s_{j+1},0}^{(j)} \\ \theta_{11}^{(j)} & \theta_{21}^{(j)} & \theta_{31}^{(j)} & \cdots & \theta_{s_{j+1},1}^{(j)} \\ \theta_{12}^{(j)} & \theta_{22}^{(j)} & \theta_{32}^{(j)} & \cdots & \theta_{s_{j+1},2}^{(j)} \\ \theta_{13}^{(j)} & \theta_{23}^{(j)} & \theta_{33}^{(j)} & \cdots & \theta_{s_{j+1},3}^{(j)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \theta_{1,s_j}^{(j)} & \theta_{2,s_j}^{(j)} & \theta_{3,s_j}^{(j)} & \cdots & \theta_{s_{j+1},s_j}^{(j)} \end{pmatrix} \\
&= \begin{pmatrix} z^{(j+1)(1)} \\ z^{(j+1)(2)} \\ z^{(j+1)(3)} \\ \vdots \\ z^{(j+1)(m)} \end{pmatrix} \\
&= \begin{pmatrix} z_1^{(j+1)(1)} & z_2^{(j+1)(1)} & z_3^{(j+1)(1)} & \cdots & z_{s_{j+1}}^{(j+1)(1)} \\ z_1^{(j+1)(2)} & z_2^{(j+1)(2)} & z_3^{(j+1)(2)} & \cdots & z_{s_{j+1}}^{(j+1)(2)} \\ z_1^{(j+1)(3)} & z_2^{(j+1)(3)} & z_3^{(j+1)(3)} & \cdots & z_{s_{j+1}}^{(j+1)(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_1^{(j+1)(m)} & z_2^{(j+1)(m)} & z_3^{(j+1)(m)} & \cdots & z_{s_{j+1}}^{(j+1)(m)} \end{pmatrix} \\
&\Rightarrow (m, s_j + 1) * (s_j + 1, s_{j+1}) = (m, s_{j+1}) \\
a^{(j+1)} &= g(z^{(j+1)}) \Rightarrow (m, s_{j+1})
\end{aligned} \tag{43}$$

### 4.2.2 y

$$y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ \vdots \\ y^{(m)} \end{pmatrix}_{m \times 1} \tag{44}$$

为进行矩阵运算，要将其转化为如下形式:<sup>注[3]</sup>

$$Y = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}_{m, s_L} \quad (45)$$

注[4]

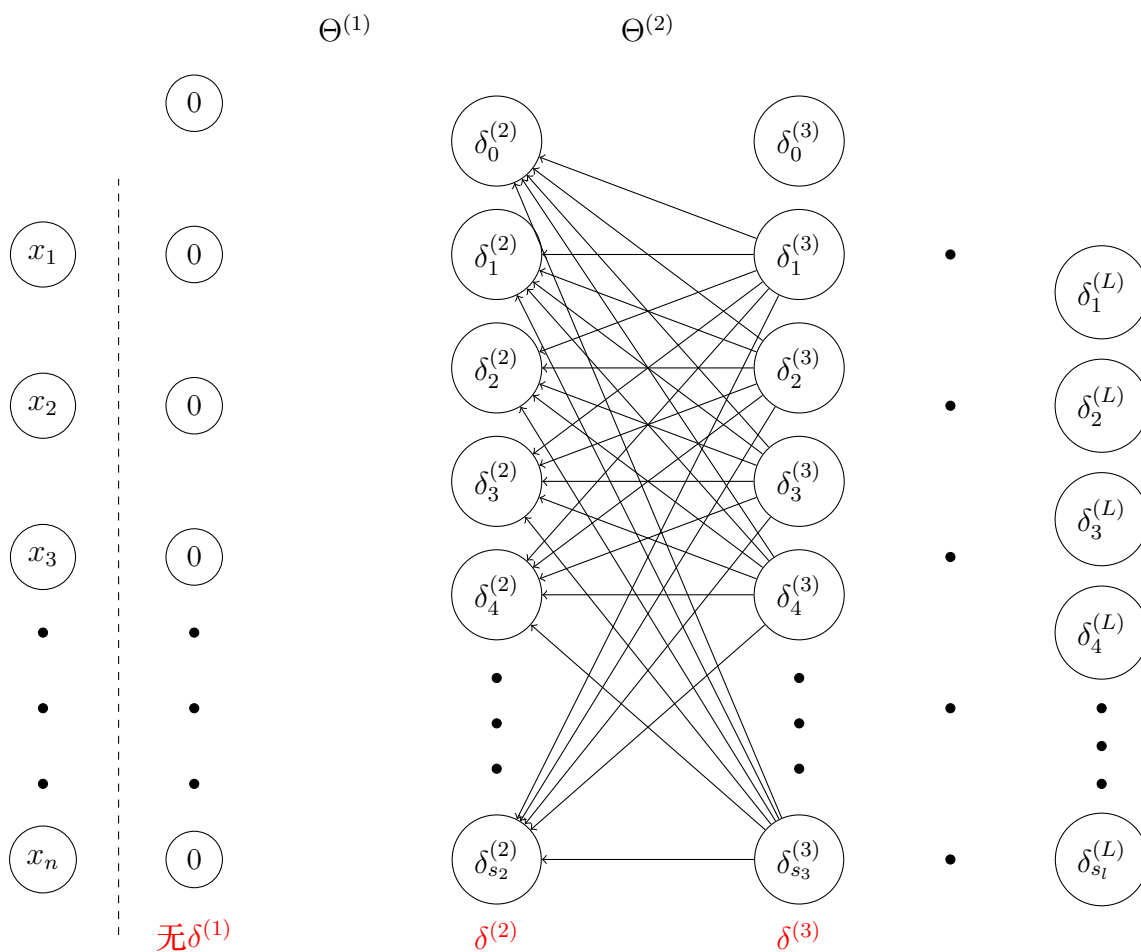
---

<sup>注[3]</sup>y所对应的值所在的索引位置值为1，其他位置均为0

<sup>注[4]</sup>上式 $m * s_L$ 中的 $s_L$ 表示共有 $s_L$ 个分类器， $s_L$ 表示的是输出层的unit数

## 5 神经网络 - 后向算法

### 5.1 神经网络示意图 - 后向算法



$$\begin{cases} \delta^L = a^L - y, & l = L \\ \delta^{L-1} = (\Theta^{(L-1)})^T \delta^L \cdot g'(z^{L-1}) \\ \quad = (\Theta^{(L-1)})^T \delta^L \cdot g(z^{L-1}) \cdot (1 - g(z^{L-1})), & l = L - 1 \\ \delta^l = (\Theta^{(l)})^T [\delta^{l+1}(2 : \text{end})] \cdot g'(z^l) \\ \quad = (\Theta^{(l)})^T [\delta^{l+1}(2 : \text{end})] \cdot g(z^l) \cdot (1 - g(z^l)), & 2 \leq l \leq L - 2 \end{cases}$$

无  $\delta^{(1)}$

$$\Delta^{(l)} := \Delta^{(l)} + \delta^{(l+1)}(a^{(l)})^T \quad (46)$$

$$D_{ij}^{(l)} = \begin{cases} \frac{1}{m} \Delta_{ij}^{(l)}, & j = 0 \\ \frac{1}{m} (\Delta_{ij}^{(l)} + \Theta_{ij}^{(l)}), & j \neq 0 \end{cases}$$

$$\frac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}} = D_{ij}^{(l)}$$

## 5.2 神经网络 - 后向算法

### 5.2.1 输出层结果: $a^L$

$$a^L = \begin{pmatrix} a_1^{(L)(1)} & a_2^{(L)(1)} & a_3^{(L)(1)} & \dots & a_{s_L}^{(L)(1)} \\ a_1^{(L)(2)} & a_2^{(L)(2)} & a_3^{(L)(2)} & \dots & a_{s_L}^{(L)(2)} \\ a_1^{(L)(3)} & a_2^{(L)(3)} & a_3^{(L)(3)} & \dots & a_{s_L}^{(L)(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1^{(L)(m)} & a_2^{(L)(m)} & a_3^{(L)(m)} & \dots & a_{s_L}^{(L)(m)} \end{pmatrix}_{m, s_L} \quad (47)$$

### 5.2.2 格式化后的Y

$$Y = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}_{m, s_L} \quad (48)$$

### 5.2.3 $\delta^L$

$$\begin{aligned} \delta^L &= a^L - y \\ &= \begin{pmatrix} a_1^{(L)} \\ a_2^{(L)} \\ a_3^{(L)} \\ \vdots \\ a_{s_L}^{(L)} \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \end{aligned} \quad (49)$$

- 此时,  $\delta^L, a^L, y$ 表示的是均向量 (不是矩阵)

### 5.2.4 $\delta^{L-1}$

$$\begin{aligned} \delta^{L-1} &= (\Theta^{(L-1)})^T \delta^L \cdot g'(z^{L-1}) \\ &= (\Theta^{(L-1)})^T \delta^L \cdot g(z^{L-1}) \cdot (1 - g(z^{L-1})) \end{aligned} \quad (50)$$

1. 其中, 式 $g'(z) = g(z)(1 - g(z))$ , 此为sigmoid函数的特性
2. 此时,  $z^{L-1}$ 表示的是一个向量 (不是矩阵)
3. 此时不需要舍弃 $\delta_0^L$ , 因为根本就没有

### 5.2.5 $\delta^l (2 \leq l \leq L - 2)$

$$\begin{aligned} \delta^l &= (\Theta^{(l)})^T [\delta^{l+1}(2 : \text{end})] \cdot g'(z^l) \\ &= (\Theta^{(l)})^T [\delta^{l+1}(2 : \text{end})] \cdot g(z^l) \cdot (1 - g(z^l)) \end{aligned} \quad (51)$$

1. 因 $a^{(1)}$ 直接从X得到, 不会有误差, 故无 $\delta^{(1)}$
2. (2:end)表示舍弃第一个数据 $\delta_0^{s_{L-1}}$  (Matlab索引从1开始)
3. 对比于从 $a^l$ 到 $a^{l+1}$ 要添加一个 $a_0^l = 1$ ; 从 $\delta^{l+1}$ 到 $\delta^l$ 要舍弃一个 $\delta_0^{l+1}$
4. 同样地, 此时 $z^l$ 表示的是一个向量 (不是矩阵)

### 5.2.6 $\Delta^l$ (用迭代的方式计算)

#### 1. 数值计算方式

$$\Delta_{ij}^{(l)} := \Delta_{ij}^{(l)} + a_j^{(l)} \delta_i^{(l+1)} \quad (52)$$

#### 2. 矩阵计算方式

$$\Delta^{(l)} := \Delta^{(l)} + \delta^{(l+1)} (a^{(l)})^T \quad (53)$$

### 5.2.7 $D_{ij}^{(l)}$

#### 1. $j = 0$ 时

$$D_{ij}^{(l)} := \frac{1}{m} \Delta_{ij}^{(l)} \quad (54)$$

#### 2. $j \neq 0$ 时

$$D_{ij}^{(l)} := \frac{1}{m} (\Delta_{ij}^{(l)} + \Theta_{ij}^{(l)}) \quad (55)$$

### 5.2.8 $\frac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}}$

$$\frac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}} = D_{ij}^{(l)} \quad (56)$$

### 5.2.9 $\delta^l$ 与 $\Delta^l$ 的区别与联系