Machine Learning Formulas

MingShun Wu

September 12, 2016

Contents

1 线性回归(Linear Regression)

1.1 当训练集X只有1项时

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

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

$$y = y \tag{3}$$

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

1.2 当训练集X有m项时

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

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

$$y = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ \vdots \\ y^{(m)} \end{pmatrix}_{m*1} \tag{7}$$

1.2.1 Cost Function

1. 数值形式:

$$J(\theta) = \frac{1}{2m}\left[h_\theta(x^{(i)}) - y^{(i)}\right]^2 \tag{8}$$

2. 矩阵形式:

$$J(\theta) = \frac{1}{2m}\left[h_\theta(x) - y\right]^T\left[h_\theta(x) - y\right] \tag{9}$$

1.2.2 梯度下降

1. 数值形式

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\left[h_\theta(x^{(i)}) - y^{(i)}\right]x_j^{(i)} \tag{10}$$

迭代方式:

$$\theta_j := \theta_j - \alpha\frac{1}{m}\left[h_\theta(x^{(i)}) - y^{(i)}\right]x_j^{(i)} \tag{11}$$

2. 矩阵形式

$$\nabla J(\theta) = \frac{1}{2m}X^T\left[h_\theta(x) - y\right] \tag{12}$$

迭代方式:

$$\theta := \theta - \alpha\frac{1}{m}X^T\left[h_\theta(x) - y\right] \tag{13}$$

1.3 Feature Normalization

$$x_i = \frac{x_i - \mu}{\sigma} \tag{14}$$

或

$$x_i = \frac{x_i - \mu}{max - min} \tag{15}$$

1.4 公式法求解(Normal Equation)

$$\theta = (X^TX)^{-1}X^Ty \tag{16}$$

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

$$h_\theta(x) = \frac{1}{1+e^{\theta^T x}} \tag{17}$$

2. 矩阵形式

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

2.1.2 Cost Function

1. 数值形式

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}logh_\theta(x^{(i)}) - (1-y^{(i)})log(1-h_\theta(x^{(i)}))\right] \tag{19}$$

2. 矩阵形式

$$J(\theta) = \frac{1}{m}\left[-y^T \log h_\theta(x) - (1-y^T)\log\left(1-h_\theta(x)\right]\right. \tag{20}$$

2.1.3 梯度下降

1. 数值形式

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left[h_\theta(x^{(i)}) - y^{(i)}\right]x_j^{(i)} \tag{21}$$

迭代方式:

$$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left[h_\theta(x^{(i)}) - y^{(i)}\right]x_j^{(i)} \tag{22}$$

2. 梯矩阵形式

$$\nabla J(\theta) = \frac{1}{m}X^T\left[h_\theta(x) - y\right] \tag{23}$$

迭代方式:

$$\theta := \theta - \alpha\frac{1}{m}X^T\left[h_\theta(x) - y\right] \tag{24}$$

2.2 当只有k个类别时，使用k个分类器

2.3 避免过拟合

1. 线性回归

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}[h_\theta(x^{(i)}) - y^{(i)}]^2 + \lambda\frac{1}{2m}\sum_{j=1}^{n}\theta_j^2 \tag{25}$$

$$\theta_j = \theta_j(1-\alpha\frac{\lambda}{m}) - \frac{\alpha}{m}\sum_{i=1}^{n}\left[h_\theta(x) - y\right]x_j^{(i)} \tag{26}$$

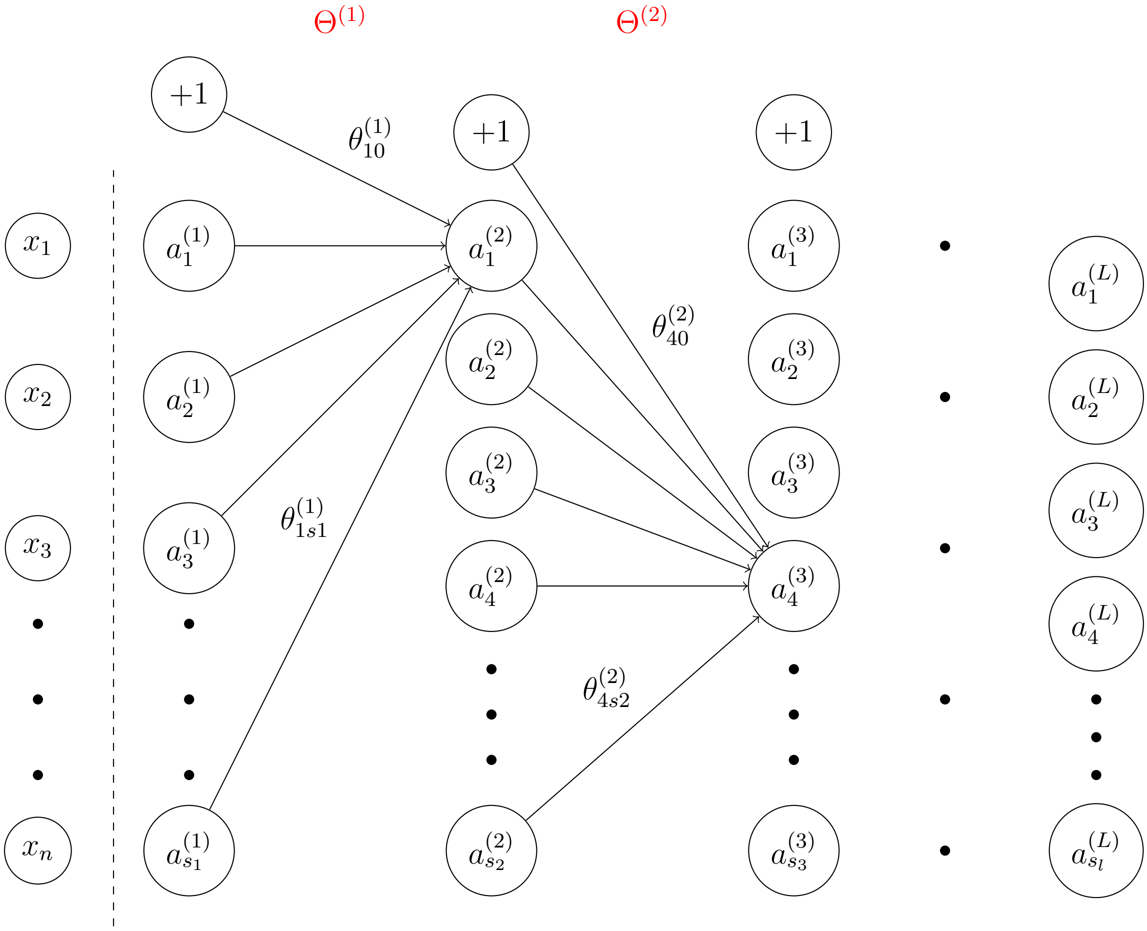
2. 逻辑回归

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}logh_\theta(x^{(i)}) - (1-y^{(i)})log(1-h_\theta(x^{(i)}))\right] + \lambda\frac{1}{2m}\sum_{j=1}^{n}\theta_j^2 \tag{27}$$

$$\theta_j = \theta_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}[h_\theta(x^{(i)}) - y^{(i)}]x_j^{(i)} + \frac{\lambda}{m}\theta_j\right] \tag{28}$$

3 神经网络

3.1 神经网络示意图



3.2 神经网络－前向算法

3.2.1 X、θ、Θ、z、a

1. X

$$
\begin{aligned}
X &= \begin{pmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ (x^{(3)})^T \\ \vdots \\ (x^{(m)})^T \end{pmatrix} \\
&= \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \dots & x_n^{(2)} \\ x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & \dots & x_n^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \\
&= \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \dots & x_n^{(2)} \\ x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & \dots & x_n^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \dots & x_n^{(m)} \end{pmatrix} \Rightarrow (m, n)
\end{aligned} \tag{29}
$$

2. $a^{(1)}$

$$
a^{(1)} = X \Rightarrow (m, n) \tag{30}
$$

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

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

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}\\
&= \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}
\tag{32}
$$

注[2]

5. $a^{(2)}$

$$a^{(2)} = g(z^{(2)}) \Rightarrow (m,s_2) \tag{33}$$

6. 后续同理

$$
\begin{aligned}
\Theta^{(2)} &= \begin{pmatrix} \theta_{10}^{(2)} & \theta_{11}^{(2)} & \theta_{12}^{(2)} & \dots & \theta_{1,s_2}^{(2)}\\ \theta_{20}^{(2)} & \theta_{21}^{(2)} & \theta_{22}^{(2)} & \dots & \theta_{2,s_2}^{(2)}\\ \theta_{30}^{(2)} & \theta_{31}^{(2)} & \theta_{32}^{(2)} & \dots & \theta_{3,s_2}^{(2)}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \theta_{s_30}^{(2)} & \theta_{s_31}^{(2)} & \theta_{s_32}^{(2)} & \dots & \theta_{s_3,s_2}^{(2)} \end{pmatrix} \Rightarrow (s_3, s_2+1)\\
z^{(3)} &= (1,a^{(2)})(\Theta^{(2)})^T \Rightarrow (m,s_2+1)*(s_2+1,s_3) = (m,s_3)\\
a^{(3)} &= g(z^{(3)}) \Rightarrow (m,s_3)\\
&\vdots
\end{aligned}
\tag{34}
$$

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)} & \dots & \theta_{1,s_j}^{(j)} \\
\theta_{20}^{(j)} & \theta_{21}^{(j)} & \theta_{22}^{(j)} & \dots & \theta_{2,s_j}^{(j)} \\
\theta_{30}^{(j)} & \theta_{31}^{(j)} & \theta_{32}^{(j)} & \dots & \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)} & \dots & \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)} & \dots & a_{s_j}^{(j)(1)} \\
1 & a_1^{(j)(2)} & a_2^{(j)(2)} & a_3^{(j)(2)} & \dots & a_{s_j}^{(j)(2)} \\
1 & a_1^{(j)(3)} & a_2^{(j)(3)} & a_3^{(j)(3)} & \dots & a_{s_j}^{(j)(3)} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
1 & a_1^{(j)(m)} & a_2^{(j)(m)} & a_3^{(j)(m)} & \dots & a_{s_j}^{(j)(m)}
\end{pmatrix}
\begin{pmatrix}
\theta_{10}^{(j)} & \theta_{20}^{(j)} & \theta_{30}^{(j)} & \dots & \theta_{s_{j+1},0}^{(j)} \\
\theta_{11}^{(j)} & \theta_{21}^{(j)} & \theta_{31}^{(j)} & \dots & \theta_{s_{j+1},1}^{(j)} \\
\theta_{12}^{(j)} & \theta_{22}^{(j)} & \theta_{32}^{(j)} & \dots & \theta_{s_{j+1},2}^{(j)} \\
\theta_{13}^{(j)} & \theta_{23}^{(j)} & \theta_{33}^{(j)} & \dots & \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)} & \dots & \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)} & \dots & z_{s_{j+1}}^{(j+1)(1)} \\
z_1^{(j+1)(2)} & z_2^{(j+1)(2)} & z_3^{(j+1)(2)} & \dots & z_{s_{j+1}}^{(j+1)(2)} \\
z_1^{(j+1)(3)} & z_2^{(j+1)(3)} & z_3^{(j+1)(3)} & \dots & 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)} & \dots & 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{35}
$$

3.2.2 y

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

为进行矩阵运算，要将其转化为如下形式:注[3]

$$Y = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}_{m,s_L} \tag{37}$$

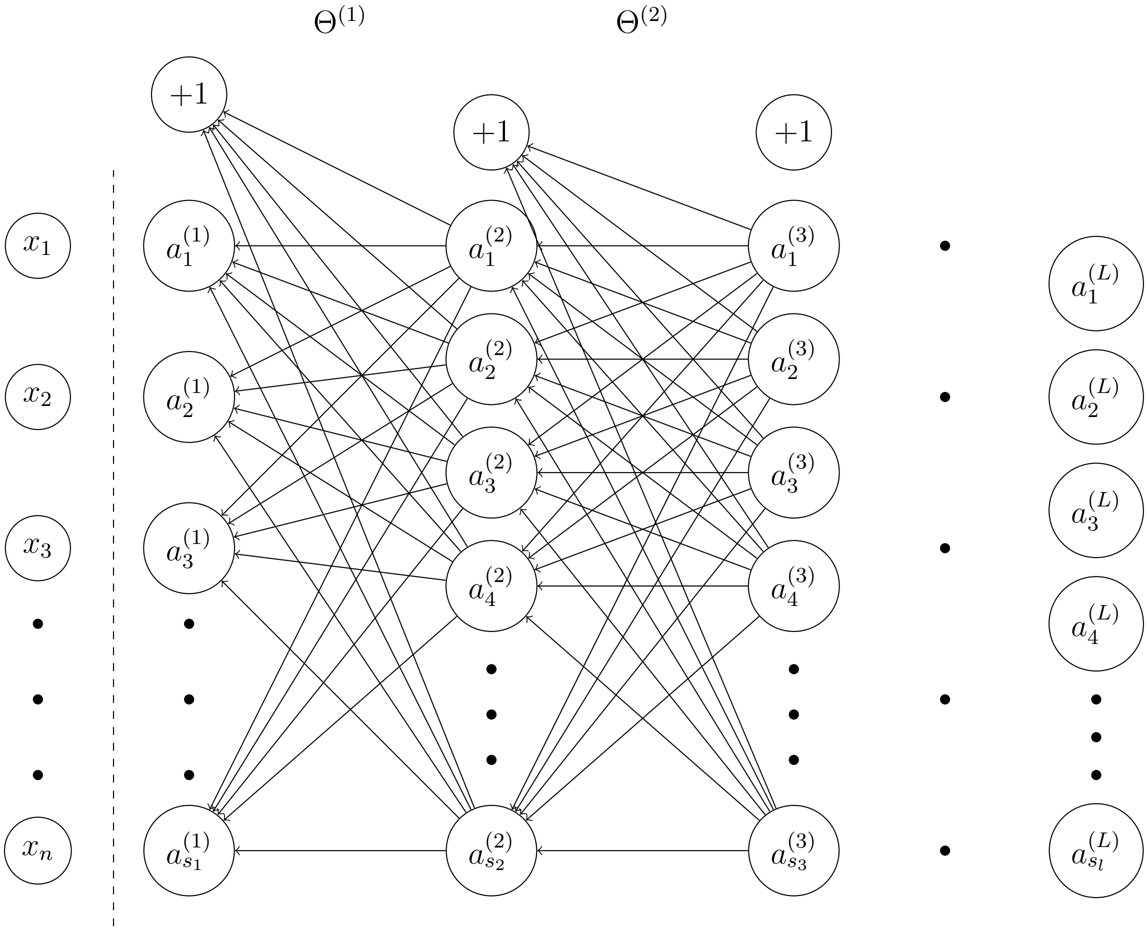
注[4]

注[3]y所对应的值所在的索引位置值为1，其他位置均为0

注[4]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

4 神经网络

4.1 神经网络示意图 – 后向算法



$$\delta^{(1)} = (\Theta^{(1)})^T[\delta^2(2:end)].*g(z^1).*(1-g(z^1))$$

$$\delta^{(l)} = (\Theta^{(l)})^T[\delta^{l+1}(2:end)].*g(z^l).*(1-g(z^l))$$

4.2 神经网络－后向算法

4.2.1 输出层结果：a^L

$$a^L = \begin{pmatrix} a_1^{(L)(1)} & a_2^{(L)(1)} & a_3^{(L)(1)} & \dots & a_{s_L}^{(L)(1)} \\ a_1^{(L)(2)} & a_2^{(L)(2)} & a_3^{(L)(2)} & \dots & a_{s_L}^{(L)(2)} \\ a_1^{(L)(3)} & a_2^{(L)(3)} & a_3^{(L)(3)} & \dots & a_{s_L}^{(L)(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1^{(L)(m)} & a_2^{(L)(m)} & a_3^{(L)(m)} & \dots & a_{s_L}^{(L)(m)} \end{pmatrix}_{m,s_L} \tag{38}$$

4.2.2 格式化后的Y

$$Y = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}_{m,s_L} \tag{39}$$

4.2.3 δ^L

$$\begin{aligned} \delta^L &= a^L - y \\ &= \begin{pmatrix} a_1^{(L)} \\ a_2^{(L)} \\ a_3^{(L)} \\ \vdots \\ a_{s_L}^{(L)} \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \end{aligned} \tag{40}$$

- 此时，δ^L, a^L, y表示的是均向量（不是矩阵）

4.2.4 δ^{L-1}

$$\begin{aligned} \delta^{L-1} &= (\Theta^{(L-1)})^T \delta^L .* g'(z^{L-1}) \\ &= (\Theta^{(L-1)})^T \delta^L .* g(z^{L-1}) .* (1 - g(z^{L-1})) \end{aligned} \tag{41}$$

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δ_0^L，因为根本就没有

4.2.5 δ^l （$2 <= l <= L-2$）

$$\begin{aligned} \delta^l &= (\Theta^{(l)})^T [\delta^{l+1}(2:end)] .* g'(z^l) \\ &= (\Theta^{(l)})^T [\delta^{l+1}(2:end)] .* g(z^l) .* (1 - g(z^l)) \end{aligned} \tag{42}$$

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δ^{l+1}到δ^l要舍弃一个δ_0^{l+1}
4. 同样地，此时z^l表示的是一个向量（不是矩阵）

4.2.6 Δ^l（用迭代的方式计算）

1. 数值计算方式

$$\Delta_{ij}^{(l)} := \Delta_{ij}^{(l)} + a_j^{(l)}\delta_i^{(l+1)} \tag{43}$$

2. 矩阵计算方式

$$\Delta^{(l)} := \Delta^{(l)} + \delta^{(l+1)}(a^{(l)})^T \tag{44}$$

4.2.7 $D_{ij}^{(l)}$

1. $j = 0$时

$$D_{ij}^{(l)} := \frac{1}{m}\Delta_{ij}^{(l)} \tag{45}$$

2. $j \neq 0$时

$$D_{ij}^{(l)} := \frac{1}{m}(\Delta_{ij}^{(l)} + \Theta_{ij}^{(l)}) \tag{46}$$

4.2.8 $\frac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}}$

$$\frac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}} = D_{ij}^{(l)} \tag{47}$$

4.2.9 δ^l 与 Δ^l的区别与联系