神经网络算法

By Xian2207, 13689903575, wszhangxian@126.com

目录

5.1 神经元模型

答：神经元模型是神经网络最基本的成分。神经网络定义是具有自适应性的简单单元组成的广泛并行互联的网络，该网络模拟生物神经系统对外进行交互。简单单元即神经元模型。常见的模型如下：M-P 神经元模型（也称阈值逻辑单元），M 和 P 分别指人名的缩写字母，该模型主要组成为：输入参数，连接权值，阈值，激活函数（y）。

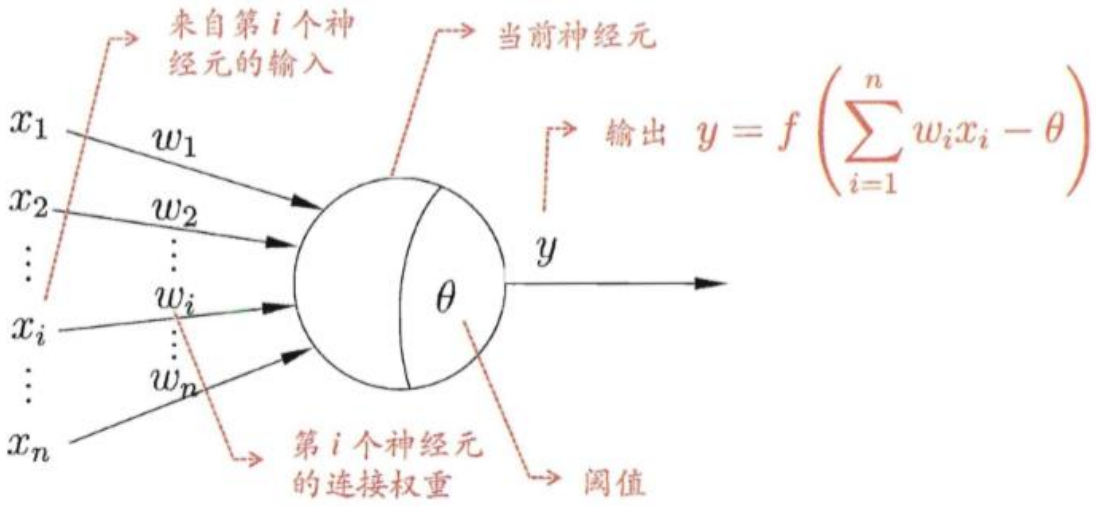
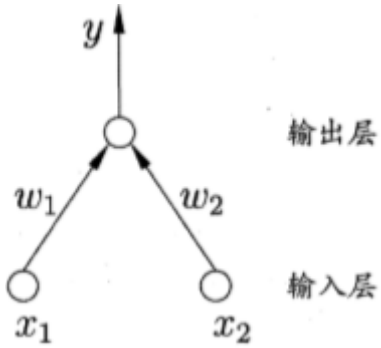


图 5.1 M-P 神经元模型

5.2 感知机和多层网络

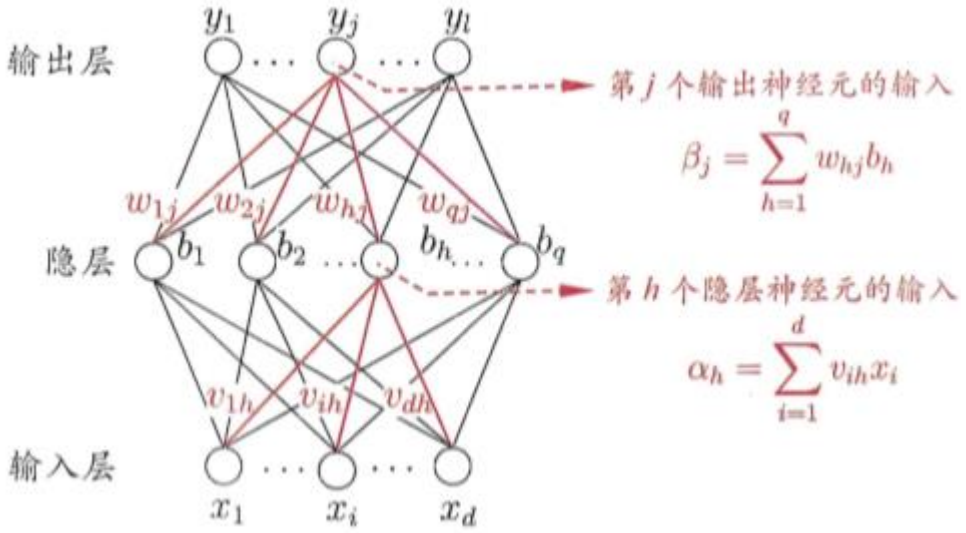
答：感知机由两层神经元组成，输入层是输入参数，输出层是 M-P 神经元。如图



感知机只有一层进行激活函数处理，故感知机也被称为功能神经元，学习能力非常有限。后提出隐层或隐含层，如果是多层隐层，则为多层前馈神经网络。此类感知机隐层和输出层都有激活函数处理。一般情况下，两层感知机被称为单隐层网络。神经网络的学习过程就是调整层与层之间的连接权值和神经元的阈值。

5.3 误差逆传播算法

答：即 Back Propagation algorithm，简称 BP 算法。给定训练集 D={(x1,y1), (x2,y2), ..., (xi,yi), (xd, yd)}，显然输入层是 x1, x2, ..., xi, xd 共 d 个属性，输出层是 y1, y2, ..., yl 个，因为输出层个数一般小于输入层，如类的划分。连接权值和阈值的计算分别用 alpha 和 beta 表示，激活函数用 f 表示，则 BP 算法草图如下



隐层含有 b1, b2, ...bh, bq 个神经元，输出层含有 y1, y2, ..., yl 个神经元，alpha 是有属性 v 和属性值 x 乘积确定，alpha 作为输入层计算出来的值传递给隐层 b1, b2,..., bh, bq 神经元。隐层神经元计算出来的 beta 值传递给输出层 y1,y2,...,yl 神经元。 假设训练集有一样本(x_k,y_k)，神经网络输出为 y_k(y_1^k, y_2^k, y_l^k)即

$$\hat{y}_j^k = f(\beta_j - \theta_j) \ , \tag{5.3}$$

theta 为神经元的阈值，则神经网络(x_k,y_k)上的均方误差为

$$E_k = \frac{1}{2}\sum_{j=1}^{l}(\hat{y}_j^k - y_j^k)^2 \ . \tag{5.4}$$

由于神经网络训练时要计算的是连接权值和阈值，所以如何求 v 和 w 是重点

$$v \leftarrow v + \Delta v \ . \tag{5.5}$$

更新权值需借助梯度下降算法，给定学习率 η , 权值步长值表示为

$$\Delta w_{hj} = -\eta \frac{\partial E_k}{\partial w_{hj}} \ . \tag{5.6}$$

根据链式法则，输出层 y 基于 beta, beta 基于 whj, 所以导数变为

$$\frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}} \ . \tag{5.7}$$

根据 beta 定义，

$$\frac{\partial \beta_j}{\partial w_{hj}} = b_h \ . \tag{5.8}$$

激活函数 f(x)求导

$$f'(x) = f(x)(1 - f(x)) \ , \tag{5.9}$$

根据 5.3 和 5.4

$$\begin{aligned} g_j &= -\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \\ &= -(\hat{y}_j^k - y_j^k) f'(\beta_j - \theta_j) \\ &= \hat{y}_j^k (1 - \hat{y}_j^k)(y_j^k - \hat{y}_j^k) \ . \end{aligned} \tag{5.10}$$

将 5.10 和 5.8 带入 5.7，再代入 5.6 则有

$$\Delta w_{hj} = \eta g_j b_h \ . \tag{5.11}$$

类似可得

$$\Delta \theta_j = -\eta g_j \ , \tag{5.12}$$

$$\Delta v_{ih} = \eta e_h x_i \ , \tag{5.13}$$

$$\Delta \gamma_h = -\eta e_h \ , \tag{5.14}$$

其中 e_h 为

$$
\begin{aligned}
e_h &= -\frac{\partial E_k}{\partial b_h} \cdot \frac{\partial b_h}{\partial \alpha_h} \\
&= -\sum_{j=1}^{l} \frac{\partial E_k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial b_h} f'(\alpha_h - \gamma_h)
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{j=1}^{l} w_{hj} g_j f'(\alpha_h - \gamma_h) \\
&= b_h(1-b_h) \sum_{j=1}^{l} w_{hj} g_j \ . \qquad (5.15)
\end{aligned}
$$

算法流程

输入： 训练集 $D = \{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^{m}$;
学习率 η.

过程：

1: 在$(0,1)$范围内随机初始化网络中所有连接权和阈值
2: **repeat**
3:　**for all** $(\boldsymbol{x}_k, \boldsymbol{y}_k) \in D$ **do**
4:　　根据当前参数和式(5.3) 计算当前样本的输出 $\hat{\boldsymbol{y}}_k$;
5:　　根据式(5.10) 计算输出层神经元的梯度项 g_j;
6:　　根据式(5.15) 计算隐层神经元的梯度项 e_h;
7:　　根据式(5.11)-(5.14) 更新连接权 w_{hj}, v_{ih} 与阈值 θ_j, γ_h
8:　**end for**
9: **until** 达到停止条件

输出： 连接权与阈值确定的多层前馈神经网络

图 5.8　误差逆传播算法

但要注意，BP 算法终极目标是最小化训练集 D 上的累积误差

$$
E = \frac{1}{m} \sum_{k=1}^{m} E_k \ , \qquad (5.16)
$$

如果 E_k 用 5.4 式推导，则推导出的是标准 BP 算法；如果用 5.16 式推导，则得出累积误差逆传播算法。前者计算迭代次数多，但能得到较好的解；后者迭代次数少，但解到一定程度。

精度难以提高。一般都是配合使用，如训练集较大时，先后者再前者。由于 BP 算法表示能力强大，常出现过拟合，故用正则化法或早停法解决过拟合。正则化法的误差公式为

$$E = \lambda \frac{1}{m} \sum_{k=1}^{m} E_k + (1-\lambda) \sum_i w_i^2 , \qquad (5.17)$$

早停法是将数据集分成训练集合验证集，训练集用来计算梯度，更新连接权值，阈值；验证集用来估计误差，若训练集误差降低但验证集误差升高，则停止训练，同时返回最小验证集误差的连接权和阈值。

5.4 全局最小和局部最小

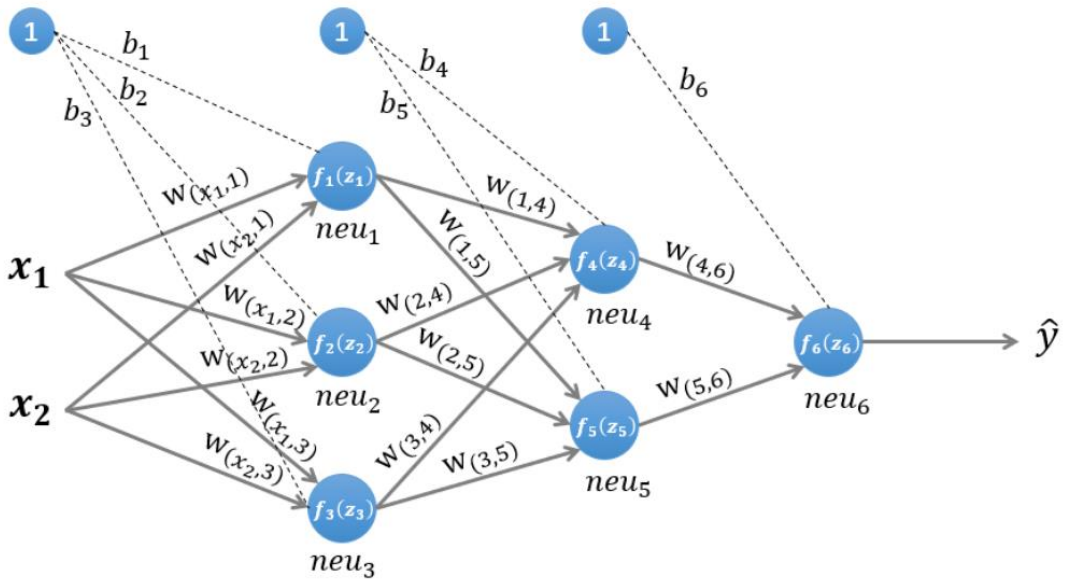
答：即最优解问题。主要算法是牛顿家族算法，遗传算法（PSO，DE，GA，模拟退火）等。

5.5 常用其他神经网络

答：RBF 网络（Radial Basis Function），ART 网络（Adaptive Resonance Theory），SOM 网络（Self-Organizing Map），级联相关网络（Cascade-Correlation），Elman 网络，Boltzman 机（energy-based model），深度学习（CNN 和 DBN-Deep Belief Network）.

5.6 举例

假设网络有三层，如下图，实施 BP 算法。



第一层神经元：neu_1，neu_2，neu_3

第二层神经元：neu_4，neu_5
第三层神经元：neu_6

1 前向传播计算

输入层：

$$\vec{a} = (x1, x2)$$

第一层网络参数：

$$W^1 = \begin{bmatrix} w_{(x1,1)} & w_{(x2,1)} \\ w_{(x2,2)} & w_{(x2,2)} \\ w_{(x1,3)} & w_{(x2,3)} \end{bmatrix}, b^1 = [b1, b2, b3]$$

第二层网络参数：

$$W^2 = \begin{bmatrix} w_{(1,4)} & w_{(2,4)} & w_{(3,4)} \\ w_{(1,5)} & w_{(2,5)} & w_{(3,5)} \end{bmatrix}, b^2 = [b4, b5]$$

第三层网络参数：

$$W^3 = [w_{(4,6)} \quad w_{(5,6)}], b^3 = [b6]$$

1.1 第一个隐层

$$z = W \times \vec{\alpha} + \vec{b}$$

根据上式，第一个隐层的三个神经元计算值为

$$z^1 = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = W^1 \times \vec{\alpha} + b^1$$

则

$$z_1 = w_{(x1,1)} \cdot x_1 + w_{(x2,1)} \cdot x_2 + b_1$$
$$z_2 = w_{(x1,2)} \cdot x_1 + w_{(x2,2)} \cdot x_2 + b_2$$
$$z_3 = w_{(x1,3)} \cdot x_1 + w_{(x2,3)} \cdot x_2 + b_3$$

假设选择$f(x)$作为该层的激活函数，同一层激活函数一般都是一样的，不同层可选择不同的激活函数，那么该层对应的输出记为

$$f(z^1) = \begin{bmatrix} f_1(z_1) \\ f_2(z_2) \\ f_3(z_3) \end{bmatrix}$$

1.2 第二个隐层

第二个隐层只有两个神经元，输入为第一层的输出，即

$$z^2 = \begin{bmatrix} z_4 \\ z_5 \end{bmatrix} = W^2 \cdot f(z^1) + b^2$$

则

$$z^2 = \begin{bmatrix} z_4 \\ z_5 \end{bmatrix} = \begin{bmatrix} w_{(1,4)} & w_{(2,4)} & w_{(3,4)} \\ w_{(1,5)} & w_{(2,5)} & w_{(3,5)} \end{bmatrix} \cdot (z_1, z_2, z_3) + \begin{bmatrix} b_4 \\ b_5 \end{bmatrix}$$

该输出层为

$$f(z^2) = \begin{bmatrix} f_4(z_4) \\ f_5(z5) \end{bmatrix}$$

1.3 输出层

输出层就一个神经元

$$z^3 = z_6 = W^3 \cdot f(z^2) + b^3$$

则

$$z^3 = z_6 = [w_{(4,6)} \quad w_{(5,6)}] \cdot (z_4, z_5) + b_6$$

输出层使用同一个激活函数，则最后输出为

$$f(x) = f_6(z^3)$$

2 反向传播计算

假设使用随机梯度下降方式来学习神经网络的参数，损失函数定义为$L(y, \overline{y})$，其中y是该样本的真实类标。要使用梯度下降，必须求解连接权 w 和偏置 b 的偏导数。由于$L(y, \overline{y})$和y的计算是前者依赖后者，后者一拉 w 和 b，根据链式法则有

$$\frac{\partial L(y, \overline{y})}{\partial W^k} = \frac{\partial L(y, \overline{y})}{\partial z^k} \cdot \frac{\partial z^k}{\partial W^k}$$

$$\frac{\partial L(y, \overline{y})}{\partial b^k} = \frac{\partial L(y, \overline{y})}{\partial z^k} \cdot \frac{\partial z^k}{\partial b^k}$$

故只需求$\frac{\partial L(y,\overline{y})}{\partial z^k}$，$\frac{\partial z^k}{\partial W^k}$，$\frac{\partial z^k}{\partial b^k}$ 这三项

2.1 偏导数$\frac{\partial z^k}{\partial b^k}$

因偏置 b 是一个常数项，偏导数为

$$\frac{\partial z^k}{\partial b^k} = \begin{bmatrix} \frac{\partial(W_1^k \cdot n^{k-1} + b_1)}{\partial b^1}, \frac{\partial(W_1^k \cdot n^{k-1} + b_2)}{\partial b^2}, \cdots, \frac{\partial(W_1^k \cdot n^{k-1} + b_m)}{\partial b^m} \\ \frac{\partial(W_2^k \cdot n^{k-1} + b_1)}{\partial b^1}, \frac{\partial(W_2^k \cdot n^{k-1} + b_2)}{\partial b^2}, \cdots, \frac{\partial(W_2^k \cdot n^{k-1} + b_m)}{\partial b^m} \\ \vdots \\ \frac{\partial(W_m^k \cdot n^{k-1} + b_1)}{\partial b^1}, \frac{\partial(W_m^k \cdot n^{k-1} + b_2)}{\partial b^2}, \cdots, \frac{\partial(W_m^k \cdot n^{k-1} + b_m)}{\partial b^m} \end{bmatrix}$$

其中

$$n^k = f_k(z^k)$$

例如，第一层

$$\frac{\partial z^1}{\partial b^1}=\begin{bmatrix}\frac{\partial(z_1)}{\partial b^1},\frac{\partial(z_1)}{\partial b^2},\frac{\partial(z_1)}{\partial b^3}\\ \frac{\partial(z_2)}{\partial b^1},\frac{\partial(z_2)}{\partial b^2},\frac{\partial(z_2)}{\partial b^3}\\ \frac{\partial(z_3)}{\partial b^1},\frac{\partial(z_3)}{\partial b^2},\frac{\partial(z_3)}{\partial b^3}\end{bmatrix}$$

$$\frac{\partial z^1}{\partial b^1}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

第二层

$$\frac{\partial z^2}{\partial b^2}=\begin{bmatrix}\frac{\partial z_4}{\partial b_4},\frac{\partial z_4}{\partial b_5}\\ \frac{\partial z_5}{\partial b_4},\frac{\partial z_5}{\partial b_5}\end{bmatrix}$$

$$\frac{\partial z^2}{\partial b^2}=\begin{bmatrix}1&0\\0&1\end{bmatrix}$$

第三层

$$\frac{\partial z^3}{\partial b^3}=\begin{bmatrix}\frac{\partial z_6}{\partial b_6}\end{bmatrix}$$

$$\frac{\partial z^3}{\partial b^3}=1$$

2.2 偏导数$\frac{\partial L(y,\overline{y})}{\partial z^k}$，$\frac{\partial L(y,\overline{y})}{\partial W^k}$，$\frac{\partial L(y,\overline{y})}{\partial b^k}$

偏导数$\frac{\partial L(y,\overline{y})}{\partial z^k}$被称为误差项（error term，也被称为灵敏度），一般用δ表示。例如，$\frac{\partial L(y,\overline{y})}{\partial z^1}$则表示第一层神经元的误差项，其值大小代表第一层神经元对最终误差值的影响大小。根据前一节前向计算，我们知道第$k+1$层的输入与第k层的输出关系为：

$$z^{k+1}=W^{k+1}\cdot n^k+b^k$$

因为$n^k=f_k(z^k)$，根据链式法则

$$\delta^k=\frac{\partial L(y,\overline{y})}{\partial z^k}=\frac{\partial n^k}{\partial z^k}\cdot\frac{\partial z^{k+1}}{\partial n^k}\cdot\frac{\partial L(y,\overline{y})}{\partial z^{k+1}}$$
$$=f_k'(z^k)\cdot[(W^{k+1})^T\cdot\delta^{k+1}]$$

第k层神经元的误差项δ^k是第$k+1$层的误差项δ^{k+1}与$k+1$层的权值与第k层激活函数的倒数（梯度）得到的，由此推出

$$\frac{\partial L(y,\overline{y})}{\partial W^k}=\frac{\partial L(y,\overline{y})}{\partial z^k}\cdot\frac{\partial z^k}{\partial W^k}=\delta^k\cdot(n^{k-1})^T$$

$$\frac{\partial L(y,\overline{y})}{\partial b^k}=\frac{\partial L(y,\overline{y})}{\partial z^k}\cdot\frac{\partial z^k}{\partial b^k}=\delta^k$$

3 实际数据

3.1 参数初始值

依然使用上图，假设其真实类标为 1，所有参数初始值如下

$$\vec{a} = (x1, x2) = (1,2)$$

$$W^1 = \begin{bmatrix} w_{(x1,1)} & w_{(x2,1)} \\ w_{(x2,2)} & w_{(x2,2)} \\ w_{(x1,3)} & w_{(x2,3)} \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 3 & 2 \end{bmatrix}, b^1 = [b1, b2, b3] = [1,2,3]$$

$$W^2 = \begin{bmatrix} w_{(1,4)} & w_{(2,4)} & w_{(3,4)} \\ w_{(1,5)} & w_{(2,5)} & w_{(3,5)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 2 \\ 3 & 2 & 2 \end{bmatrix}, b^2 = [b4, b5] = [2,1]$$

$$W^3 = [w_{(4,6)} \quad w_{(5,6)}] = [1,3], \quad b^3 = [b6] = 2$$

假设所有的激活函数是：

$$f(x) = \frac{1}{1 + e^{-x}}$$

使用均差函数作为损失函数：

$$L(y, \overline{y}) = \frac{1}{2} \cdot (y - \overline{y})^2$$

3.2 前向传播

3.2.1 第一层

$$z = W \times \vec{\alpha} + \vec{b}$$

$\vec{\alpha}$作为第一层的输入，计算z^1

$$z^1 = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 3 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 9 \\ 10 \end{bmatrix}$$

z^1作为 $f(x)$的输入，计算$f(z^1)$

$$f(z^1) = \begin{bmatrix} f_1(z_1) \\ f_2(z_2) \\ f_3(z_3) \end{bmatrix} = \begin{bmatrix} \frac{1}{1 + e^{-5}} \\ \frac{1}{1 + e^{-9}} \\ \frac{1}{1 + e^{-10}} \end{bmatrix} = \begin{bmatrix} 0.99331 \\ 0.99988 \\ 0.99995 \end{bmatrix}$$

3.2.2 第二层

$$z^2 = \begin{bmatrix} z_4 \\ z_5 \end{bmatrix} = W^2 \cdot f(z^1) + b^2$$

$f(z^1)$作为第二层的输入，计算z^2

$$z^2 = \begin{bmatrix} z_4 \\ z_5 \end{bmatrix} = W^2 \cdot \begin{bmatrix} f_1(z_1) \\ f_2(z_2) \\ f_3(z_3) \end{bmatrix} + b^2 = \begin{bmatrix} 1 & 1 & 2 \\ 3 & 2 & 2 \end{bmatrix} \cdot \begin{bmatrix} 0.99331 \\ 0.99988 \\ 0.99995 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 5.9931 \\ 7.9796 \end{bmatrix}$$

z^2作为 $f(x)$的输入，计算$f(z^2)$

$$f(z^2) = \begin{bmatrix} f_4(z_4) \\ f_5(z_5) \end{bmatrix} = \begin{bmatrix} \dfrac{1}{1+e^{-5.9931}} \\ \dfrac{1}{1+e^{-7.9796}} \end{bmatrix} = \begin{bmatrix} 0.99751 \\ 0.99966 \end{bmatrix}$$

3.2.3 输出层

$$z^3 = [z_6] = W^3 \cdot f(z^2) + b^3$$

$f(z^2)$作为输出层的输入，计算z^3

$$z^3 = [z_6] = [1\ 3] \cdot \begin{bmatrix} 0.99751 \\ 0.99966 \end{bmatrix} + 2 = 5.9965$$

z^3作为 $f(x)$的输入，计算$f(z^3)$

$$f(z^3) = [f_6(z_6)] = \left[\frac{1}{1+e^{-5.9965}}\right] = 0.99752$$

3.3 后向传播

3.3.1 输出层误差

$$L(y, \overline{y}) = \frac{1}{2} \cdot (y - \overline{y})^2$$

其中$y = 1$为真实类标，预测值$\overline{y} = 0.99752$，则

$$\delta^3 = \frac{\partial L(y, \overline{y})}{\partial z^k} = \frac{\partial L(y, \overline{y})}{\partial \overline{y}} \cdot \frac{\partial \overline{y}}{\partial z^k} = -(y - \overline{y}) \cdot [\overline{y} \cdot (1 - \overline{y})]$$

注意$\overline{y} = f(z^3)$，而 sigmoid 函数

$$f(x) = \frac{1}{1+e^{-x}}$$

倒数恰好是

$$f'(x) = f(x) \cdot (1 - f(x))$$

故

$$\delta^3 = -\big(1 - f(z^3)\big) \cdot \big[f(z^3) \cdot \big(1 - f(z^3)\big)\big]$$

$$\delta^3 = -(1 - 0.99752) \cdot [0.99752 \cdot (1 - 0.99752)]$$

$$\delta^3 = -0.00248 \cdot [0.0024738]$$
$$\delta^3 = 6.135 \times 10^{-6}$$

根据误差推导公式

$$\delta^k = f_k'(z^k) \cdot [(W^{k+1})^T \cdot \delta^{k+1}]$$

3.3.2 第二层误差$\boldsymbol{\delta^2}$

$$\delta^2 = f_2'(z^2) \cdot [(W^{2+1})^T \cdot \delta^{2+1}]$$

其中

$$f_2'(z^2) = \begin{bmatrix} f'(z_4) & 0 \\ 0 & f'(z_5) \end{bmatrix}$$
$$= \begin{bmatrix} f(z_4) \cdot (1 - f(z_4)) & 0 \\ 0 & f(z_5) \cdot (1 - f(z_5)) \end{bmatrix}$$
$$= \begin{bmatrix} 0.99751 \cdot (1 - 0.99751) & 0 \\ 0 & 0.99966 \cdot (1 - 0.99966) \end{bmatrix}$$
$$= \begin{bmatrix} 0.0024838 & 0 \\ 0 & 0.00033988 \end{bmatrix}$$

则

$$\delta^2 = \begin{bmatrix} 0.0024838 & 0 \\ 0 & 0.00033988 \end{bmatrix} \cdot \left[\begin{bmatrix} 1 \\ 3 \end{bmatrix} \cdot 6.135 \times 10^{-6} \right]$$
$$= \begin{bmatrix} 0.0024838 & 0 \\ 0 & 0.00033988 \end{bmatrix} \cdot \begin{bmatrix} 6.135 \times 10^{-6} \\ 1.8405 \times 10^{-5} \end{bmatrix}$$
$$= \begin{bmatrix} 1.5238 \times 10^{-8} \\ 6.2555 \times 10^{-9} \end{bmatrix}$$

3.3.3 第一层误差$\boldsymbol{\delta^1}$

$$\delta^1 = f_1'(z^1) \cdot [(W^{1+1})^T \cdot \delta^{1+1}]$$

其中

$$f_1'(z^1) = \begin{bmatrix} f'(z_1) & 0 & 0 \\ 0 & f'(z_2) & \\ 0 & 0 & f'(z_3) \end{bmatrix}$$
$$= \begin{bmatrix} 0.0066452 & 0 & 0 \\ 0 & 0.00011999 & \\ 0 & 0 & 4.9997 \times 10^{-5} \end{bmatrix}$$

则

$$\delta^1 = \begin{bmatrix} 0.0066452 & 0 & 0 \\ 0 & 0.00011999 & \\ 0 & 0 & 4.9997 \times 10^{-5} \end{bmatrix} \cdot \left(\begin{bmatrix} 1 & 3 \\ 1 & 2 \\ 2 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1.5238 \times 10^{-8} \\ 6.2555 \times 10^{-9} \end{bmatrix} \right)$$
$$= \begin{bmatrix} 2.2597 \times 10^{-10} \\ 3.3296 \times 10^{-10} \\ 2.1492 \times 10^{-12} \end{bmatrix}$$

3.4 更新权值和偏置

$$W^k = W^k - \eta \cdot (\delta^k \cdot (n^{k-1})^T - W^k)$$
$$b^k = b^k - \eta \cdot \delta^k$$

定义学习率：$\eta = 0.1$，$n^0 = \vec{\alpha}$则

3.4.1 第一层

$$W^1 = W^1 - \eta \cdot (\delta^1 \cdot (\vec{\alpha})^T - W^1)$$
$$W^1 = \begin{bmatrix}2 & 1\\ 1 & 3\\ 3 & 2\end{bmatrix} - 0.1 \cdot \left(\begin{bmatrix}2.2597 \times 10^{-10}\\ 3.3296 \times 10^{-10}\\ 2.1492 \times 10^{-12}\end{bmatrix} \cdot [1 \quad 2] - \begin{bmatrix}2 & 1\\ 1 & 3\\ 3 & 2\end{bmatrix}\right)$$
$$W^1 = \begin{bmatrix}2 & 1\\ 1 & 3\\ 3 & 2\end{bmatrix}$$
$$b^1 = \begin{bmatrix}1\\ 2\\ 3\end{bmatrix} - 0.1 \cdot \begin{bmatrix}2.2597 \times 10^{-10}\\ 3.3296 \times 10^{-10}\\ 2.1492 \times 10^{-12}\end{bmatrix} \approx \begin{bmatrix}1\\ 2\\ 3\end{bmatrix}$$

3.4.2 第二层

$$W^2 = W^2 - 0.1 \cdot (\delta^2 \cdot (n^{2-1})^T - W^2)$$

注意$n^1 = f(z^1)$，即$n^k = f_k(z^k)$，故

$$W^2 = W^2 - 0.1 \cdot (\delta^2 \cdot f(z^1)^T - W^2)$$
$$W^2 = \begin{bmatrix}1 & 1 & 2\\ 3 & 2 & 2\end{bmatrix} - 0.1 \cdot (\begin{bmatrix}1.5238 \times 10^{-8}\\ 6.2555 \times 10^{-9}\end{bmatrix} \cdot [0.99331 \quad 0.99988 \quad 0.99995] - \begin{bmatrix}1 & 1 & 2\\ 3 & 2 & 2\end{bmatrix})$$
$$W^2 = \begin{bmatrix}1.14832 & 1.4857 & 2.5858\\ 3.3 & 2.2 & 2.2\end{bmatrix}$$
$$b^2 = \begin{bmatrix}2\\ 1\end{bmatrix} - 0.1 \cdot \begin{bmatrix}1.5238 \times 10^{-8}\\ 6.2555 \times 10^{-9}\end{bmatrix} = \begin{bmatrix}2.38579\\ 1\end{bmatrix}$$

3.4.3 输出层

$$W^3 = W^3 - \eta \cdot (\delta^3 \cdot (n^{3-1})^T - W^3)$$
$$W^3 = [1,3] - 0.1 \cdot (6.135 \times 10^{-6} \cdot [0.99751\ 0.99966] - [1,3])$$
$$W^3 = [1.1\ 3.3]$$
$$b^3 = 2 - 0.1 \cdot 6.135 \times 10^{-6} \approx 2$$

至此结束，后续要看看正则化表达。