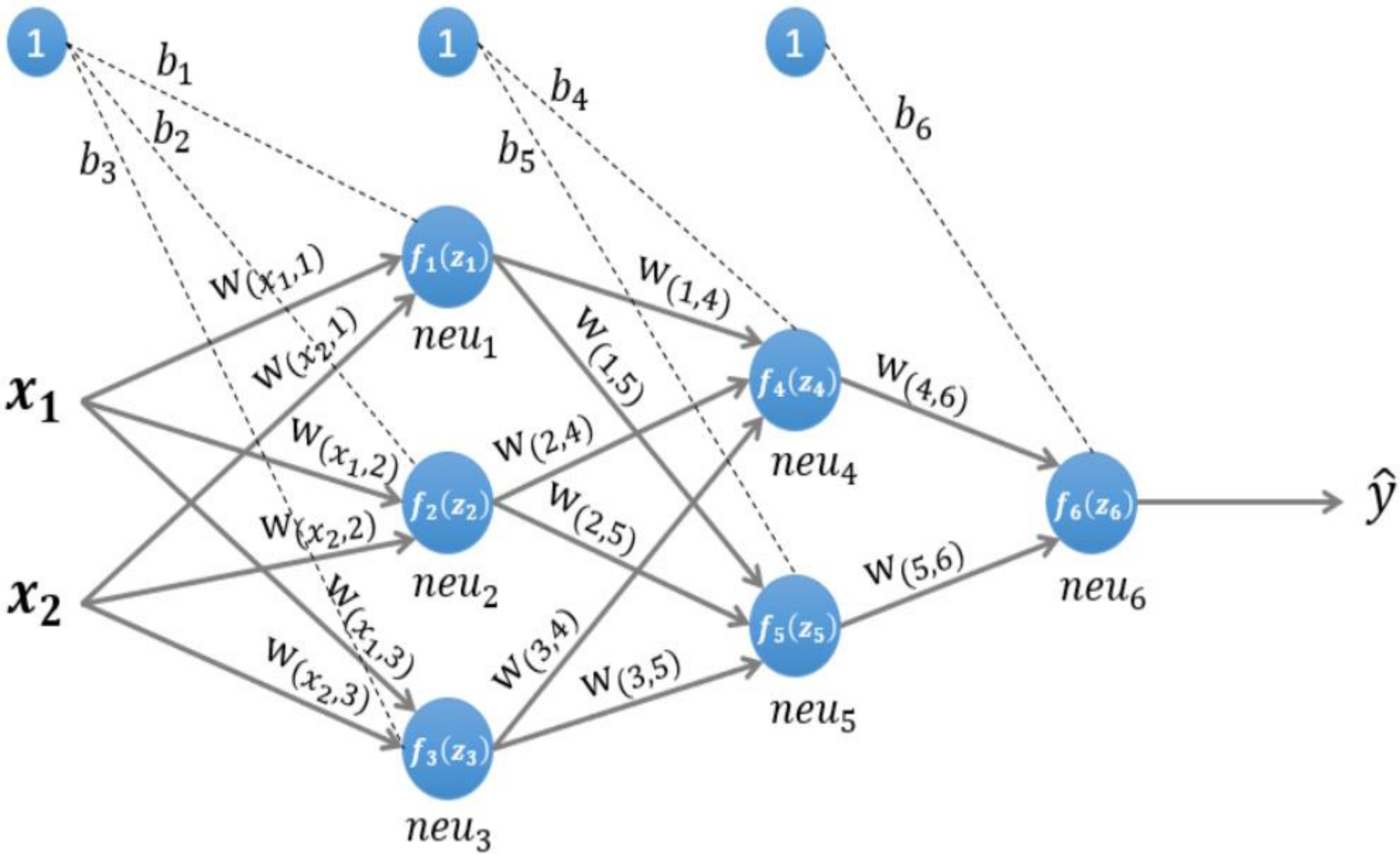
# BP算法推演

训练数据: 特征个数为2

BP网络：三层，两个隐藏层，一个输出层

分类问题：二类分类



# 1 正向传播

输入样本：$\mathbf{x} = [x_1, x_2]^T$

第一层网络参数：

$$\boldsymbol{W}^{(1)} = \begin{bmatrix} w_{(x_1,1)} & w_{(x_2,1)} \\ w_{(x_1,2)} & w_{(x_2,2)} \\ w_{(x_1,3)} & w_{(x_2,3)} \end{bmatrix}，\ b^{(1)} = [b_1 \quad b_2 \quad b_3]^T$$

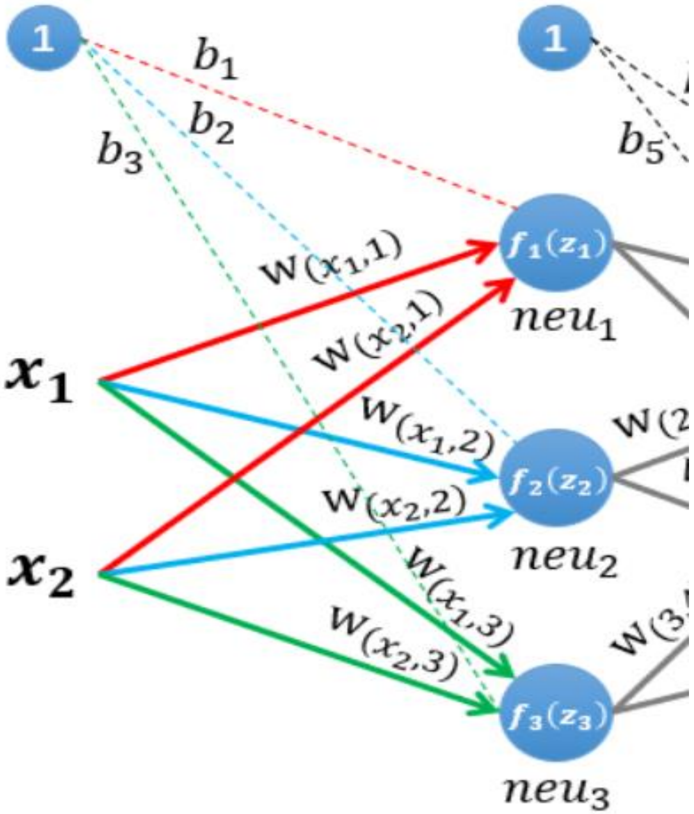
第二层网络参数：

$$\boldsymbol{W}^{(2)} = \begin{bmatrix} w_{(1,4)} & w_{(2,4)} & w_{(3,4)} \\ w_{(1,5)} & w_{(2,5)} & w_{(3,5)} \end{bmatrix}，\ b^{(2)} = [b_4 \quad b_5]^T$$

第三层网络参数：

$$\boldsymbol{W}^{(3)} = [w_{(4,6)} \quad w_{(5,6)}]，\ b^{(3)} = [b_6]^T$$

# 1.1第一层隐含层的计算




第一层有三个神经元，该层的输入为：

$$\boldsymbol{z}^{(1)} = \boldsymbol{W}^{(1)} * \mathbf{x} + \boldsymbol{b}^{(1)}$$

$neu_1$，$neu_2$，$neu_3$的输入为:

$$z_1 = w_{(x_1,1)} * x_1 + w_{(x_2,1)} * x_2 + b_1$$

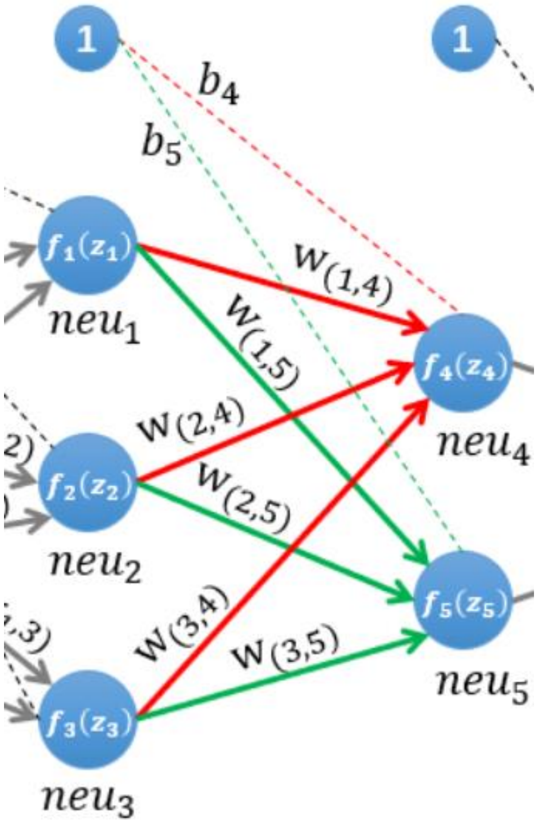
$$z_2 = w_{(x_1,2)} * x_1 + w_{(x_2,2)} * x_2 + b_2$$

$$z_3 = w_{(x_1,3)} * x_1 + w_{(x_2,3)} * x_2 + b_3$$

该层的输出为：

$$\boldsymbol{n}^{(1)} = [f_1(z_1) \quad f_2(z_2) \quad f_3(z_3)]^T$$

# 1.2第二层隐含层的计算




第二层有两个神经元，该层的输入为：

$$\boldsymbol{z}^{(2)} = \boldsymbol{W}^{(2)} * \boldsymbol{n}^{(1)} + \boldsymbol{b}^{(2)}$$
$$= \boldsymbol{W}^{(2)} * [f_1(z_1) \quad f_2(z_2) \quad f_3(z_3)]^T + \boldsymbol{b}^{(2)}$$

$neu_4$和$neu_5$的输入为:

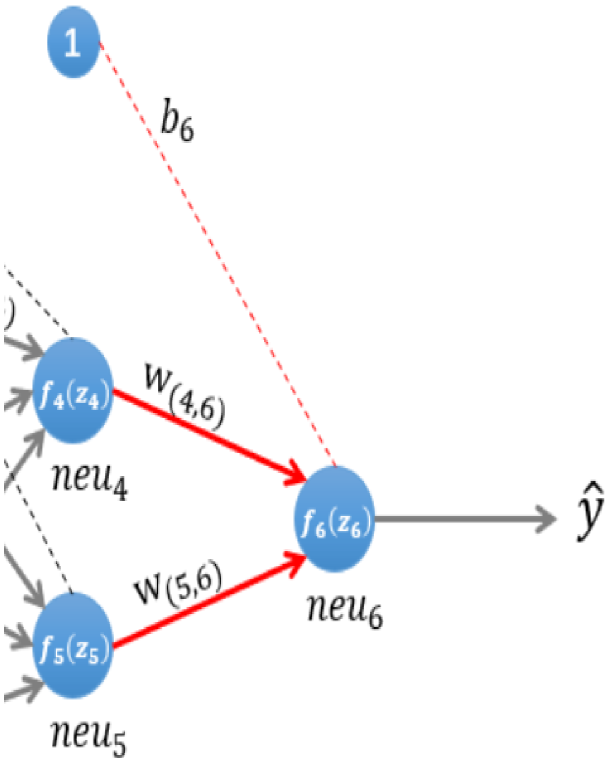
$$z_4 = w_{(1,4)} * f_1(z_1) + w_{(2,4)} * f_2(z_2) + w_{(3,4)} * f_3(z_3) + b_4$$

$$z_5 = w_{(1,5)} * f_1(z_1) + w_{(2,5)} * f_2(z_2) + w_{(3,5)} * f_3(z_3) + b_5$$

该层的输出为：

$$\boldsymbol{n}^{(2)} = [f_4(z_4) \quad f_5(z_5)\ ]^T$$

# 1.3输出层的计算




输出层只有一个神经元，该层的输入为：

$$\begin{aligned} \boldsymbol{z}^{(3)} &= \boldsymbol{W}^{(3)} * n^{(2)} + \boldsymbol{b}^{(3)} \\ &= \boldsymbol{W}^{(3)} * [f_4(z_4) \quad f_5(z_5)\ ]^T + \boldsymbol{b}^{(3)} \end{aligned}$$

$neu_6$的输入为:

$$z_6 = w_{(4,6)} * f_4(z_4) + w_{(5,6)} * f_5(z_5) + b_6$$

该层要解决二类分类问题，激活函数选Sigmoid函数，则神经网络的输出为：

$$n^{(3)} = [f_6(z_6)]^T$$

# 2 反向传播

定义损失函数：L$(y,\hat{y})$，$y$是该样本的真实类标，采用随机梯度下降法进行参数学习，计算损失函数关于神经网络各层参数（权重***W***和偏置***b***）的偏导数。

假设我们要对第k层隐藏层的参数$W^{(k)}$和$b^{(k)}$求偏导数，即求$\frac{\partial \mathrm{L}(y,\hat{y})}{\partial W^{(k)}}$和$\frac{\partial \mathrm{L}(y,\hat{y})}{\partial b^{(k)}}$。假设$z^{(k)}$代表第k层神经元的输入，即$z^{(k)} = W^{(k)} * n^{(k-1)} + b^{(k)}$，其中$n^{(k-1)}$为前一层神经元的输出，则根据链式法则有：

$$\frac{\partial \mathrm{L}(y,\hat{y})}{\partial W^{(k)}} = \frac{\partial \mathrm{L}(y,\hat{y})}{\partial z^{(k)}} * \frac{\partial z^{(k)}}{\partial W^{(k)}}$$

$$\frac{\partial \mathrm{L}(y,\hat{y})}{\partial b^{(k)}} = \frac{\partial \mathrm{L}(y,\hat{y})}{\partial z^{(k)}} * \frac{\partial z^{(k)}}{\partial b^{(k)}}$$

因此，我们只需要计算偏导数$\frac{\partial \mathrm{L}(y,\hat{y})}{\partial z^{(k)}}$、$\frac{\partial z^{(k)}}{\partial W^{(k)}}$和$\frac{\partial z^{(k)}}{\partial b^{(k)}}$。

## 2.1 计算偏导数$\frac{\partial z^{(k)}}{\partial W^{(k)}}$

根据正向传播公式，第k层的输入与第（k-1）层的输出之间的关系为：

$$\boldsymbol{z}^{(k)} = \boldsymbol{W}^{(k)} * \boldsymbol{n}^{(k-1)} + \boldsymbol{b}^{(k)}$$

则

$$\frac{\partial \boldsymbol{z}^{(k)}}{\partial \boldsymbol{W}^{(k)}} = {n^{(k-1)}}^{T}$$

## 2.2 计算偏导数 $\frac{\partial z^{(k)}}{\partial b^{(k)}}$

因为偏置 b 是一个常数项，因此偏导数的计算也很简单：

$$\frac{\partial z^{(k)}}{\partial b^{(k)}} = \begin{bmatrix} \frac{\partial(W_{1:}{}^{(k)} * n^{(k-1)} + b_1)}{\partial b_1} & \cdots & \frac{\partial(W_{1:}{}^{(k)} * n^{(k-1)} + b_1)}{\partial b_m} \\ \vdots & \cdots & \vdots \\ \frac{\partial(W_{m:}{}^{(k)} * n^{(k-1)} + b_m)}{\partial b_1} & \cdots & \frac{\partial(W_{m:}{}^{(k)} * n^{(k-1)} + b_m)}{\partial b_m} \end{bmatrix}$$

依然以第一层隐藏层的神经元为例，则有：

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

## 2.3 计算偏导数 $\frac{\partial L(y,\hat{y})}{\partial z^{(k)}}$

偏导数 $\frac{\partial L(y,\hat{y})}{\partial z^{(k)}}$ 又称为误差项(error term，也称为“灵敏度”)，一般用δ表示，例如$\delta^{(1)} = \frac{\partial L(y,\hat{y})}{\partial z^{(1)}}$ 是第一层神经元的误差项，其值的大小代表了第一层神经元对于最终总误差的影响大小。

根据第一节的前向计算，我们知道第$k+1$层的输入与第 k 层的输出之间的关系为：

$$z^{(k+1)} = W^{(k+1)} * n^{(k)} + b^{k+1}$$

又因为$n^{(k)} = f_k(z^{(k)})$，根据链式法则，我们可以得到$\delta^{(k)}$为：

$$\begin{aligned}
\delta^{(k)} &= \frac{\partial L(y,\hat{y})}{\partial z^{(k)}} \\
&= \frac{\partial n^{(k)}}{\partial z^{(k)}} * \frac{\partial z^{(k+1)}}{\partial n^{(k)}} * \frac{\partial L(y,\hat{y})}{\partial z^{(k+1)}} \\
&= \frac{\partial n^{(k)}}{\partial z^{(k)}} * \frac{\partial z^{(k+1)}}{\partial n^{(k)}} * \delta^{(k+1)} \\
&= f_k{}'\left(z^{(k)}\right) * (\left(W^{(k+1)}\right)^T * \delta^{(k+1)})
\end{aligned}$$

由上式我们可以看到，第 k 层神经元的误差项$\delta^{(k)}$是由第$k+1$层的误差项乘以第$k+1$层的权重，再乘以第 k 层激活函数的导数（梯度）得到的。这就是误差的反向传播。

## 2.4 计算 $\frac{\partial L(y,\hat{y})}{\partial W^{(k)}}$ 和 $\frac{\partial L(y,\hat{y})}{\partial b^{(k)}}$

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

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

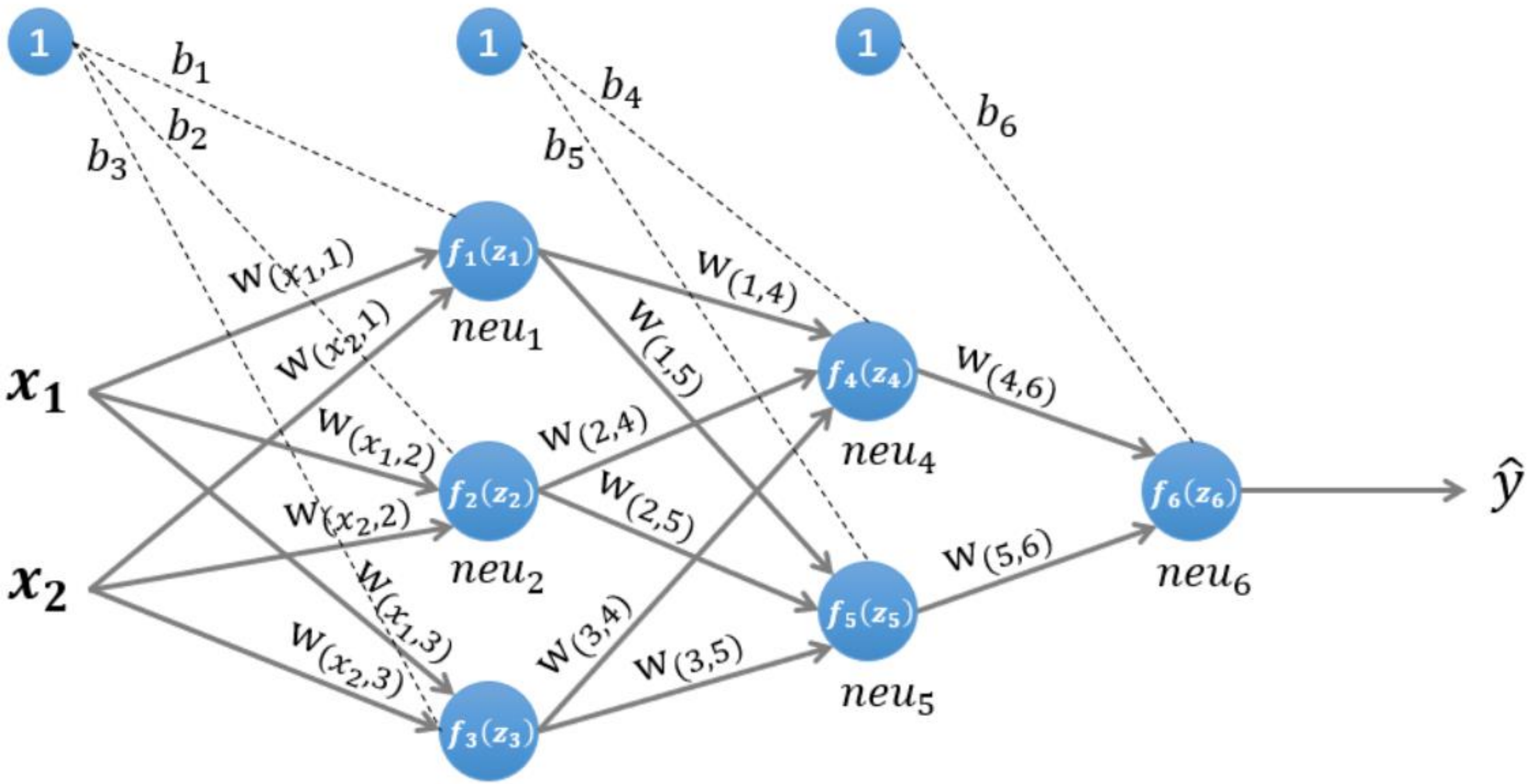
## 2.5 更新参数

$$W^{(k)} = W^{(k)} - \eta\left(\delta^{(k)}\left(n^{(k-1)}\right)^T + W^{(k)}\right)$$

$$b^{(k)} = b^{(k)} - \eta\delta^{(k)}$$

$\eta$ 为学习率

# 例子


1
1
1
$b_1$
$b_2$
$b_3$
$b_4$
$b_5$
$b_6$
$x_1$
$x_2$
$W(x_1,1)$
$W(x_2,1)$
$W(x_1,2)$
$W(x_2,2)$
$W(x_1,3)$
$W(x_2,3)$
$f_1(z_1)$
$f_2(z_2)$
$f_3(z_3)$
$neu_1$
$neu_2$
$W(1,4)$
$W(1,5)$
$W(2,4)$
$W(2,5)$
$W(3,4)$
$W(3,5)$
$f_4(z_4)$
$f_5(z_5)$
$neu_4$
$neu_5$
$W(4,6)$
$W(5,6)$
$f_6(z_6)$
$neu_6$
$\hat{y}$

例子：

输入样本：$\mathbf{x}=[x_1 \quad x_2]^T$=$[1 \quad 2]^T$，其真实类标为1

第一层网络参数：

$$W^{(1)}=\begin{bmatrix} w_{(x_1,1)}, w_{(x_2,1)} \\ w_{(x_1,2)}, w_{(x_2,2)} \\ w_{(x_1,3)}, w_{(x_2,3)} \end{bmatrix}=\begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 3 & 2 \end{bmatrix}, \qquad b^{(1)}=[b_1, b_2, b_3]^T=[1,2,3]^T$$

第二层网络参数：

$$W^{(2)}=\begin{bmatrix} w_{(1,4)}, w_{(2,4)}, w_{(3,4)} \\ w_{(1,5)}, w_{(2,5)}, w_{(3,5)} \end{bmatrix}=\begin{bmatrix} 1 & 1 & 2 \\ 3 & 2 & 2 \end{bmatrix}, \qquad b^{(2)}=[b_4, b_5]^T=[2,1]^T$$

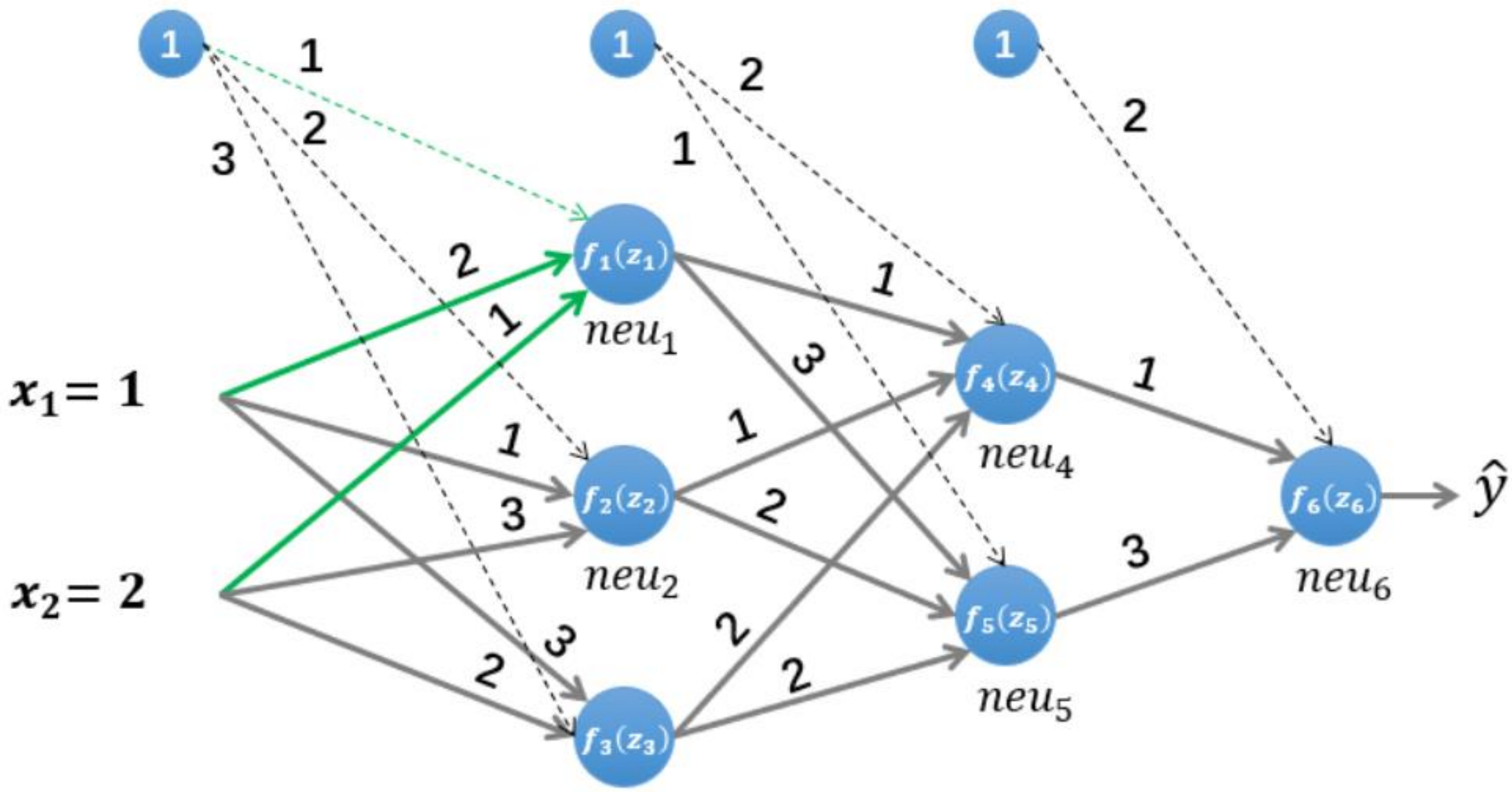
第三层网络参数：

$$W^3=\left[w_{(4,6)}, w_{(5,6)}\right]=[1,3], \qquad b^{(3)}=[b_6]=[2]$$

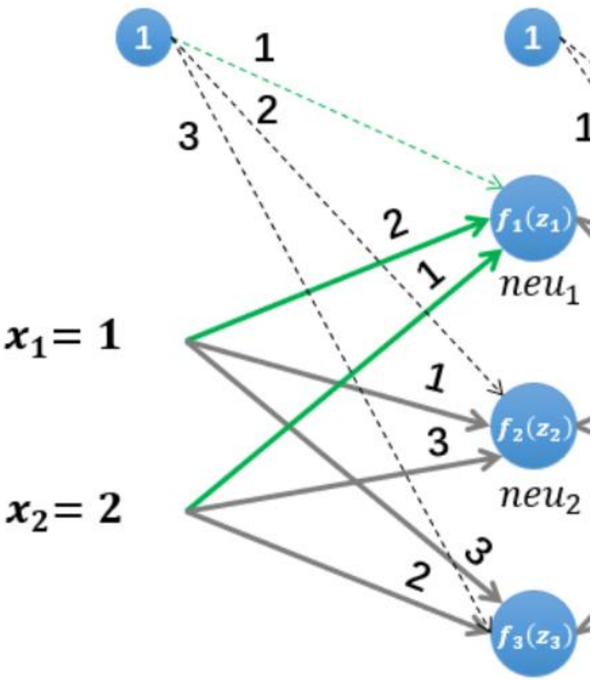
假设所有的激活函数均为 Logistic 函数：$f^{(k)}(x)=\frac{1}{1+e^{-x}}$。使用均方误差函数作为损失函数：

$$\mathrm{L}(\mathrm{y},\hat{\mathrm{y}})=\mathrm{E}(\mathrm{y}-\hat{\mathrm{y}})^2$$

为了方便求导，我们将损失函数简化为：$\mathrm{L}(\mathrm{y},\hat{\mathrm{y}})=\frac{1}{2}\sum(\mathrm{y}-\hat{\mathrm{y}})^2$。


$x_1 = 1$
$x_2 = 2$
$f_1(z_1)$
$neu_1$
$f_2(z_2)$
$neu_2$
$f_3(z_3)$
$f_4(z_4)$
$neu_4$
$f_5(z_5)$
$neu_5$
$f_6(z_6)$
$neu_6$
$\hat{y}$

# 1正向传播
## 1.1 计算第一层隐含层



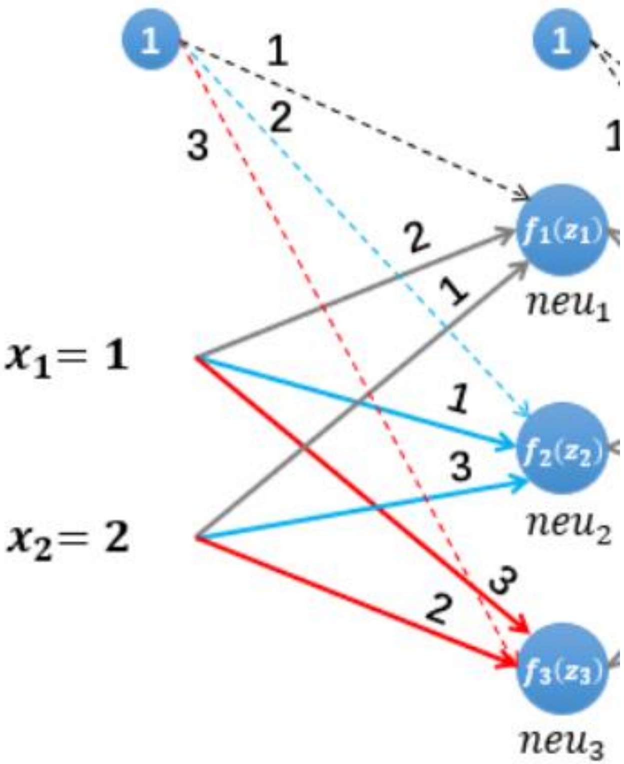

$$z_1 = w_{(x_1,1)} * x_1 + w_{(x_2,1)} * x_2 + b_1$$

$$= 2 * 1 + 1 * 2 + 1$$

$$= 5$$

$$f_1(z_1) = \frac{1}{1+e^{-z_1}} = 0.99330714907571 5$$

# 1正向传播
## 1.1 计算第一层隐含层




$$z_2 = w_{(x_1,2)} * x_1 + w_{(x_2,2)} * x_2 + b_2$$

$$= 1 * 1 + 3 * 2 + 2 = 9$$

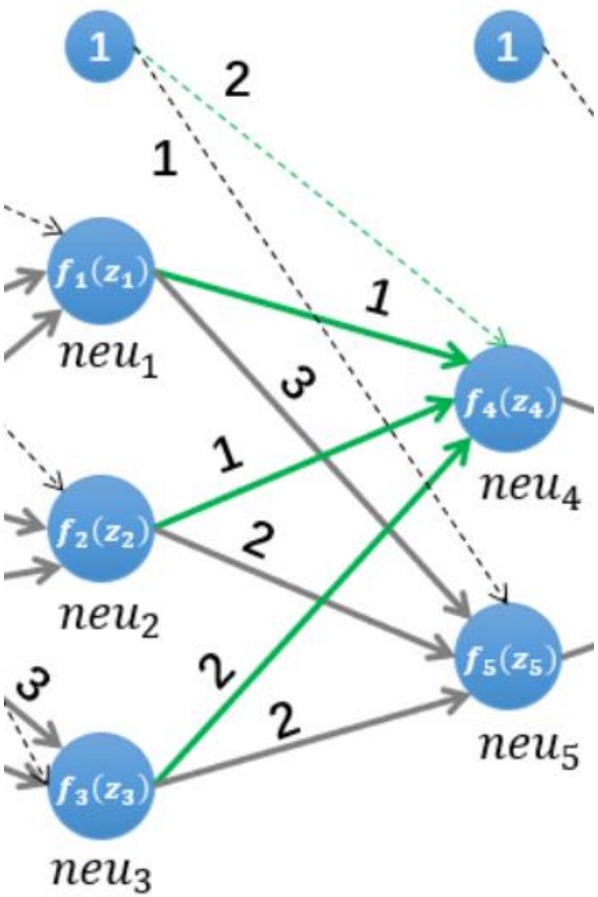
$$f_2(z_2) = \frac{1}{1 + e^{-z_2}} = 0.99987660542401 4$$

$$z_3 = w_{(x_1,3)} * x_1 + w_{(x_2,3)} * x_2 + b_3$$

$$= 3 * 1 + 2 * 2 + 3 = 10$$

$$f_3(z_3) = \frac{1}{1 + e^{-z_3}} = 0.99995460213129 8$$

# 1正向传播
## 1.2 计算第二层隐含层



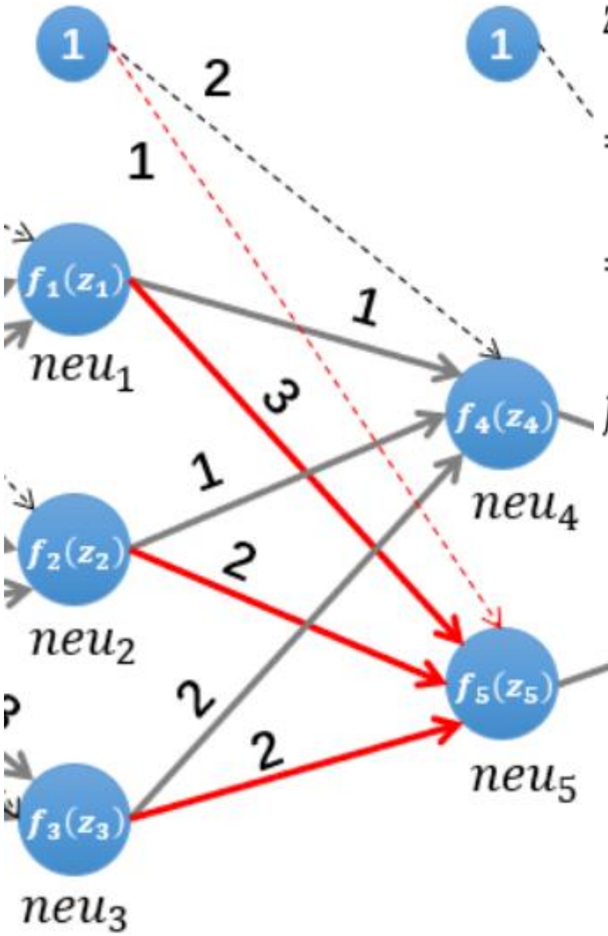

$$z_4 = w_{(1,4)} * f_1(z_1) + w_{(2,4)} * f_2(z_2) + w_{(3,4)} * f_3(z_3) + b_4$$

$$= 1 * 0.993307149075715 + 1 * 0.999876605424014 + 2 * 0.999954602131298 + 2$$

$$= 5.993092958762325$$

$$f_4(z_4) = \frac{1}{1+e^{-z_4}} = 0.997510281884102$$

# 1正向传播
## 1.2 计算第二层隐含层



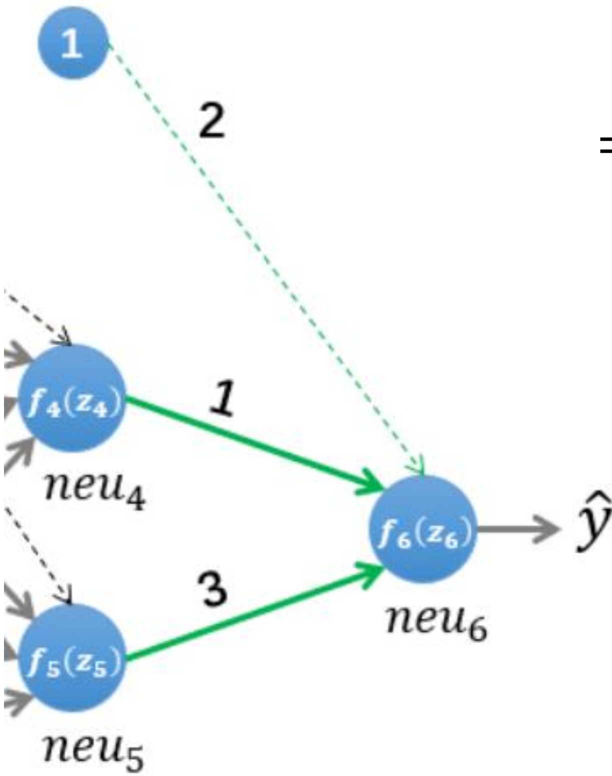

$$z_5 = w_{(1,5)} * f_1(z_1) + w_{(2,5)} * f_2(z_2) + w_{(3,5)} * f_3(z_3) + b_5$$

$$= 3 * 0.993307149075715 + 2 * 0.999876605424014 + 2 * 0.999954602131298 + 1$$

$$= 8.979460467761783$$

$$f_5(z_5) = \frac{1}{1 + e^{-z_5}} = 0.999874045072167$$

# 1正向传播
## 1.3 计算输出层




$$z_6 = w_{(4,6)} * f_4(z_4) + w_{(5,6)} * f_5(z_5) + b_6$$

=1*0.997510281884102+3*0.9998404502167+2

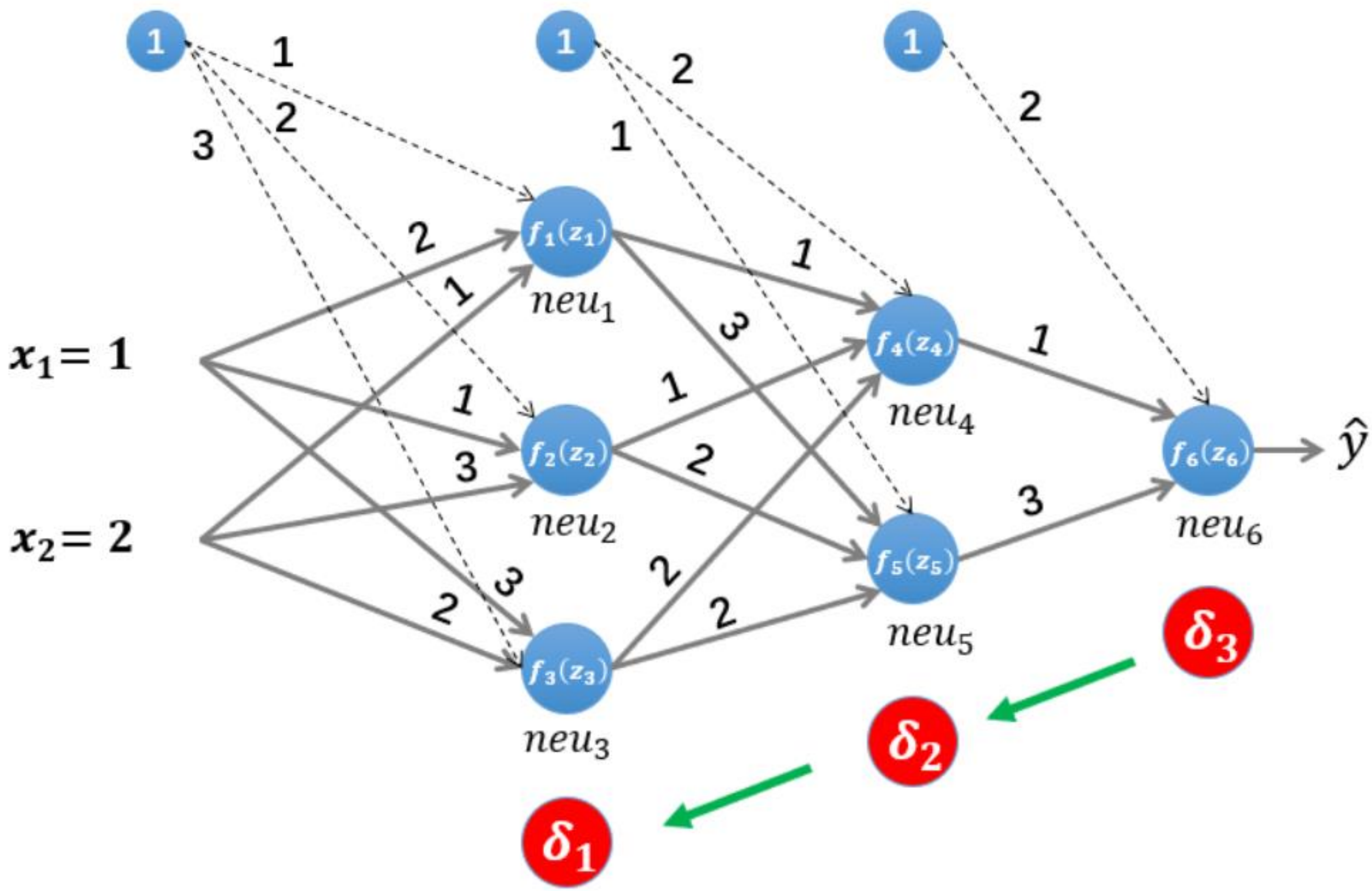
$$f_6(z_6) = \frac{1}{1 + e^{-z_6}} = 0.997520293823002$$

# 2反向传播

## 2.1 计算输网络误差

$$L=(y,\hat{y})=\frac{1}{2}\sum(y-\hat{y})^2$$

$y-\hat{y}$=1-0.997520293823002 = 0.002479706176998



# 2反向传播
## 2.1 计算输出层误差项

$$\delta_3 = \frac{\partial L(y,\hat{y})}{\partial z^{(3)}} = \frac{\partial L(y,\hat{y})}{\partial n^{(3)}} * \frac{\partial n^{(3)}}{\partial z^{(3)}} = [-0.002479706176998] * f^{(3)'}(z^3)$$

$$= [0.002473557234274] * [-0.002479706176998]$$

$$= [-0.000006133695153]$$

注：Logistic函数的求导公式

https://blog.csdn.net/qq_32768743/article/details/79118682

# 2反向传播
## 2.2 计算第二隐含层的误差项

$$\delta^{(2)} = \frac{\partial L(y,\hat{y})}{\partial z^{(2)}} = f^{(2)'}(z^{(2)}) * \left(\left(W^{(3)}\right)^{T} * \delta^{(3)}\right)$$

$$= \begin{bmatrix} f_4^{'}(z_4) & 0 \\ 0 & f_5^{'}(z_5) \end{bmatrix} * \left(\begin{bmatrix} 1 \\ 3 \end{bmatrix} * [-0.00000613369515 3]\right)$$

$$= \begin{bmatrix} 0.002483519419601 & 0 \\ 0 & 0.000125939063189 \end{bmatrix} * \begin{bmatrix} -0.000006133695153 \\ -0.000018401085459 \end{bmatrix}$$

$$= \begin{bmatrix} -0.000000015233151 \\ -0.000000002317415 \end{bmatrix}$$

# 2反向传播

## 2.3 计算第一隐含层的误差项

$$\delta^{(1)} = \frac{\partial L(y, \hat{y})}{\partial \boldsymbol{z}^{(1)}} = f^{(1)'}(\boldsymbol{z}^{(1)}) * \left( \left(W^{(2)}\right)^T * \delta^{(2)} \right)$$

$$= \begin{bmatrix} f_1{}'(z_1) & 0 & 0 \\ 0 & f_2{}'(z_2) & 0 \\ 0 & 0 & f_3{}'(z_3) \end{bmatrix} * \left( \left(W^{(2)}\right)^T * \delta^{(2)} \right)$$

$$= \begin{bmatrix} 0.006648056670790 & 0 & 0 \\ 0 & 0.00012337934 9765 & 0 \\ 0 & 0 & 0.000045395807735 \end{bmatrix}$$

$$* \left( \begin{bmatrix} 1 & 1 & 2 \\ 3 & 2 & 2 \end{bmatrix}^T * \begin{bmatrix} -0.000000015233151 \\ -0.000000002317415 \end{bmatrix} \right)$$

$$= \begin{bmatrix} -0.000000000147490 \\ -0.000000000002451 \\ -0.000000000001593 \end{bmatrix}$$

# 3更新参数

## 3.1 更新第一隐含层的参数

$$W^{(1)} = W^{(1)} - 0.1 * \left(\delta^{(1)}(n^{(0)})^T + W^{(1)}\right)$$

$$= \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 3 & 2 \end{bmatrix} - 0.1 * \left( \begin{bmatrix} -0.00000000014749 0 \\ -0.00000000000245 1 \\ -0.00000000000159 3 \end{bmatrix} * [x_1 \quad x_2] + \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 3 & 2 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 3 & 2 \end{bmatrix} - 0.1 * \left( \begin{bmatrix} -0.00000000014749 0 \\ -0.00000000000245 1 \\ -0.00000000000159 3 \end{bmatrix} * [1 \quad 2] + \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 3 & 2 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 3 & 2 \end{bmatrix} - 0.1 * \begin{bmatrix} 1.99999999985 2510 & 0.99999999970 5020 \\ 0.99999999999 7549 & 2.99999999999 5098 \\ 2.99999999999 8407 & 1.99999999999 6814 \end{bmatrix}$$

$$= \begin{bmatrix} 1.80000000001 4749 & 0.90000000002 9498 \\ 0.90000000000 0245 & 2.70000000000 0490 \\ 2.70000000000 0159 & 1.80000000000 0319 \end{bmatrix}$$

$$b^{(1)} = b^{(1)} - \eta \delta^{(1)}$$

$$= [1, 2, 3]^T - 0.1 * \begin{bmatrix} -0.00000000014749 0 \\ -0.00000000000245 1 \\ -0.00000000000159 3 \end{bmatrix}$$

$$= \begin{bmatrix} 0.99999999998 5251 \\ 1.99999999999 9755 \\ 2.99999999999 9841 \end{bmatrix}$$

# 3更新参数

## 3.2 更新第二隐含层和输出层的参数

方法同上