《新一代人工智能：从深度学习到大模型》

人工神经网络基本原理

主讲人 张重生

Part 03-1

Artificial Neural Networks

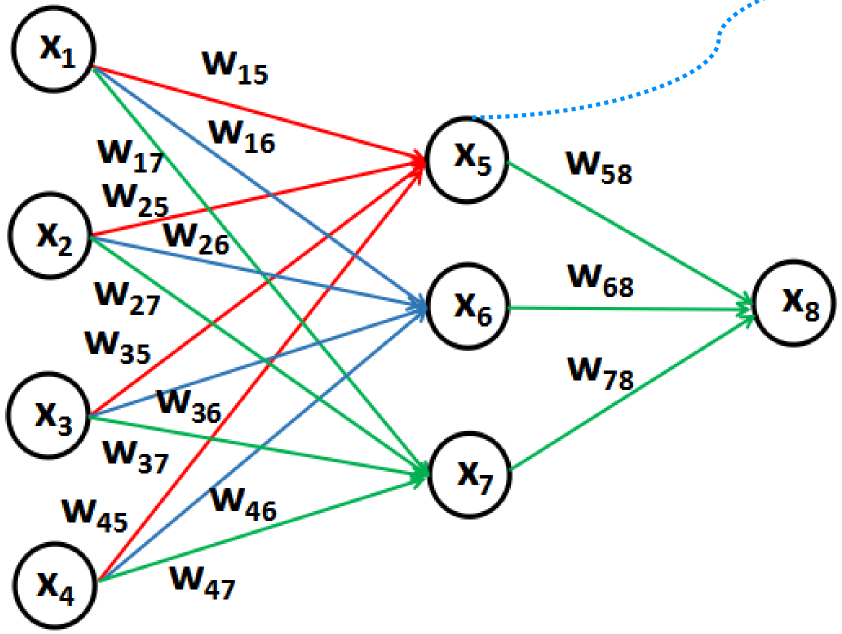
人工神经网络基本原理

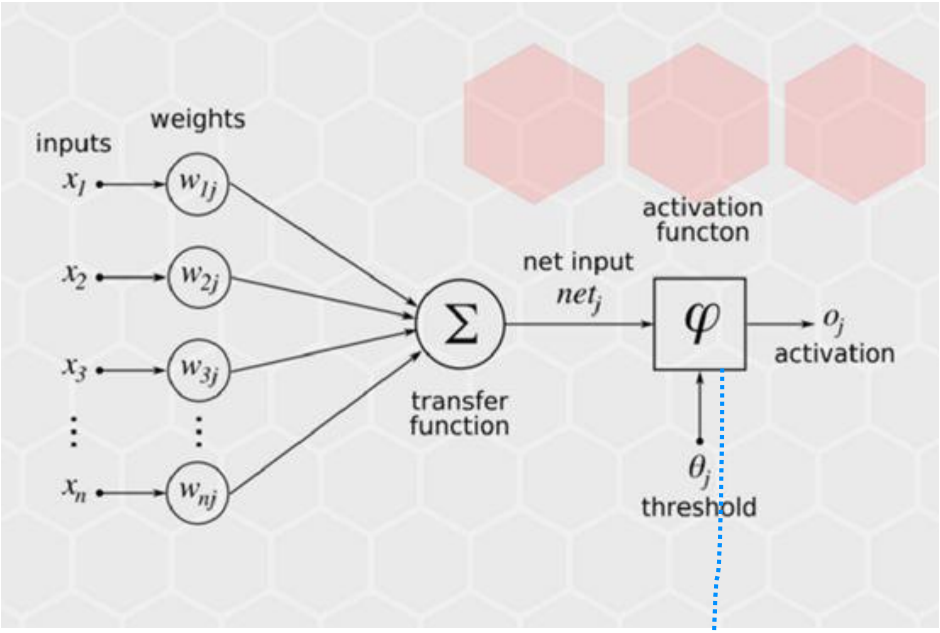
神经元、输入层、输出层、隐层的概念，神经网络中的权重；
前向传播计算

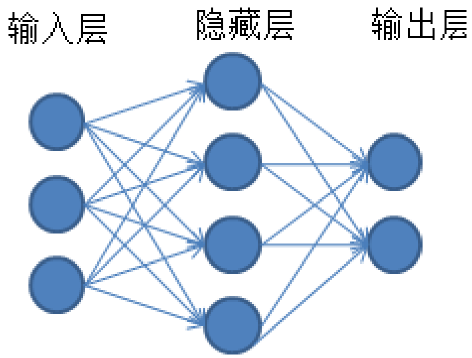
一、人工神经网络的基本概念

$f(\sum_{i=1}^{m}(w_i x_i)+b)$

X_5元素值的计算过程 $X_5 = f(w_{15}x_1+w_{25}x_2+w_{35}x_3+w_{45}x_4+b_5)$，f是激活函数







注意区分：
隐藏层激活函数
输出层激活函数

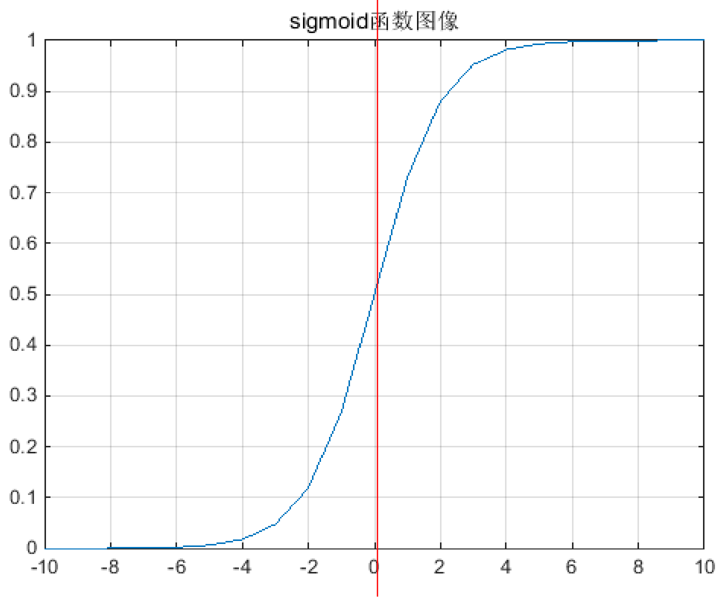
神经元，权重/权值，　　输出单元
输入层，隐藏层/隐含层，输出层

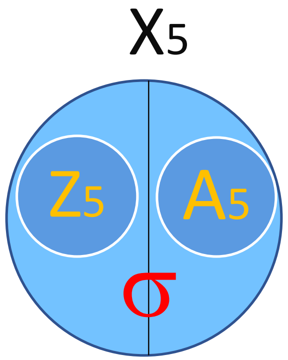
详见本书教材第40-41页

二、Sigmoid激活函数　　可以用于输出层或隐层

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

Sigmoid函数被看成是一个挤压函数，它可以将一个较大输入范围挤压到较小的0到1之间的区域。





激活前　激活后

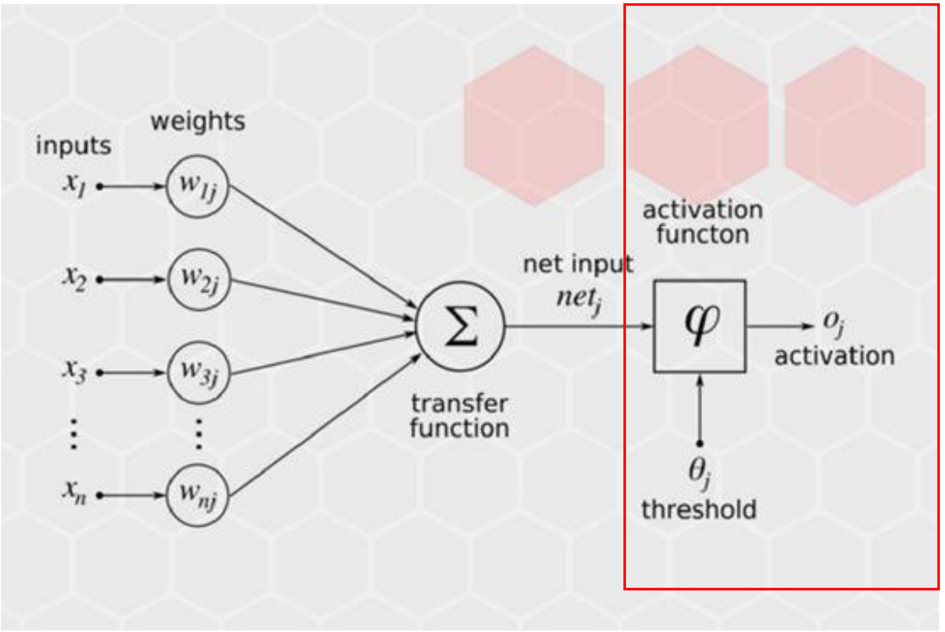
同一个神经元

Sigmoid函数求导

$$\frac{d}{dx}f(x)=\frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right)=\frac{e^{-x}}{(1+e^{-x})^2}$$

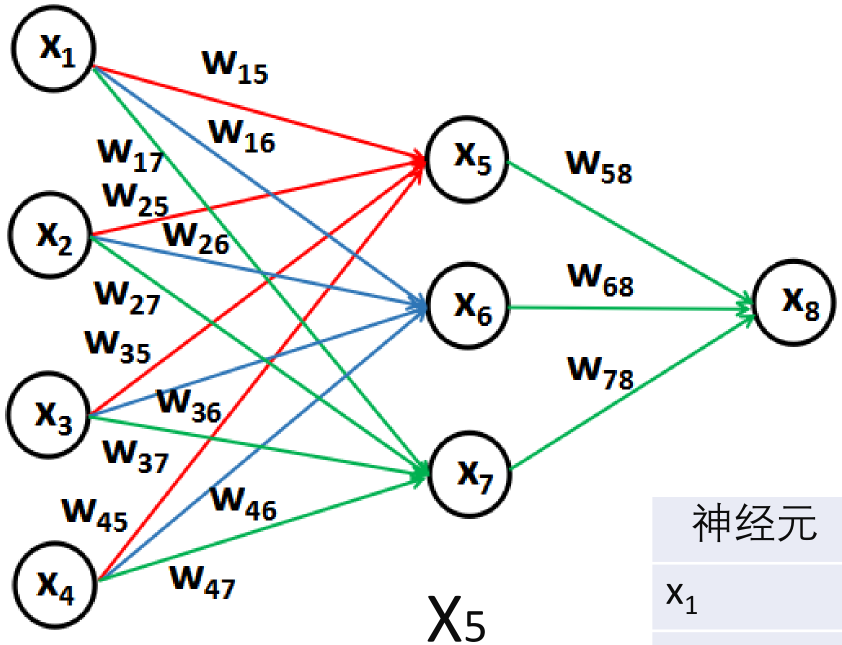
$$=f(x)*(1-f(x))$$

特点：求导容易，非线性

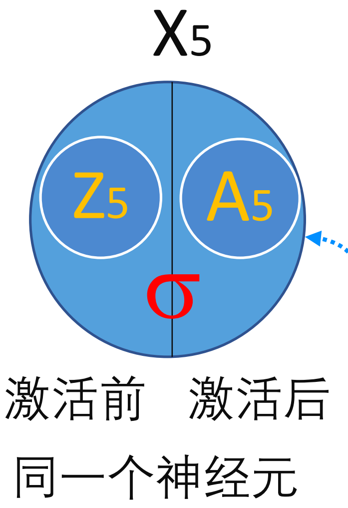


详见本书教材第41页

三、人工神经网络的前向计算



w_{15}	0.5	w_{46}	0.5
w_{16}	0.2	w_{47}	-0.1
w_{17}	-0.5	w_{58}	0.6
w_{25}	-0.2	w_{68}	-0.2
w_{26}	0.3	w_{78}	0.1
w_{27}	0.4	b_5	0.2
w_{35}	0.8	b_6	-0.5
w_{36}	-0.6	b_7	0.4
w_{37}	0.4	b_8	0.6
w_{45}	0.6		



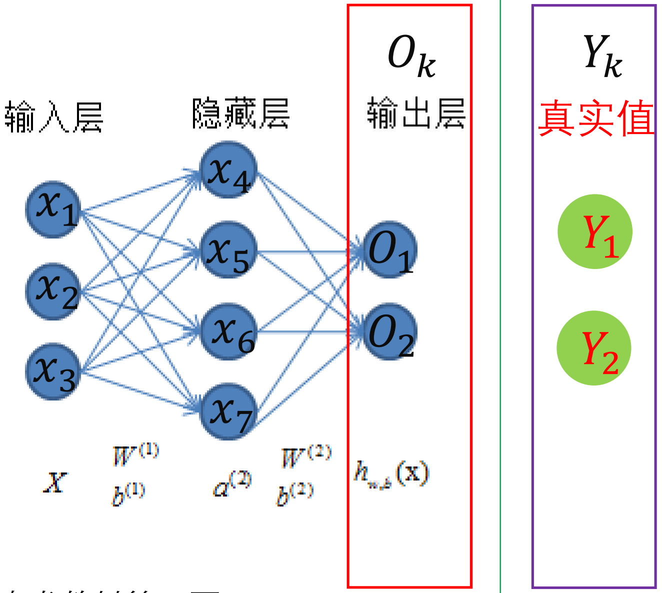
神经元	激活前	激活后
x_1	1	
x_2	0	
x_3	1	
x_4	1	
x_5	1*0.5+0*(-0.2)+1*0.8+1*0.6+0.2=2.1	$1/(1+e^{-2.1})=0.8909$
x_6	1*0.2+0*0.3+1*(-0.6)+1*0.5-0.5=-0.4	$1/(1+e^{0.4})=0.40131$
x_7	1*(-0.5)+0*0.4+1*0.4+1*(-0.1)+0.4=0.2	$1/(1+e^{-0.2})=0.54983$
x_8	0.8909*0.6+0.40131*(-0.2)+0.54983*0.1+0.6=1.1093	$1/(1+e^{-1.1093})=0.752$

详见本书教材第42-43页

四、一种简单的损失函数及其求导　　作用于输出层

$$L = \frac{1}{2}\sum_{k=1}^{C}(Y_k - O_k)^2$$

均方差损失函数
每个类别上的真实值Y_k与其预测值O_k的差的平方，最后所有类别再求和。



此时C=2
预测两个类别

详见本书教材第44页

均方差损失函数求导

$$\frac{d}{dO_k}L = \frac{d}{dO_k}\left(\frac{1}{2}\sum_{k=1}^{C}(O_k - Y_k)^2\right)$$
$$= O_k - Y_k$$

$$\frac{d}{dO_1}L = O_1 - Y_1$$
$$\frac{d}{dO_2}L = O_2 - Y_2$$

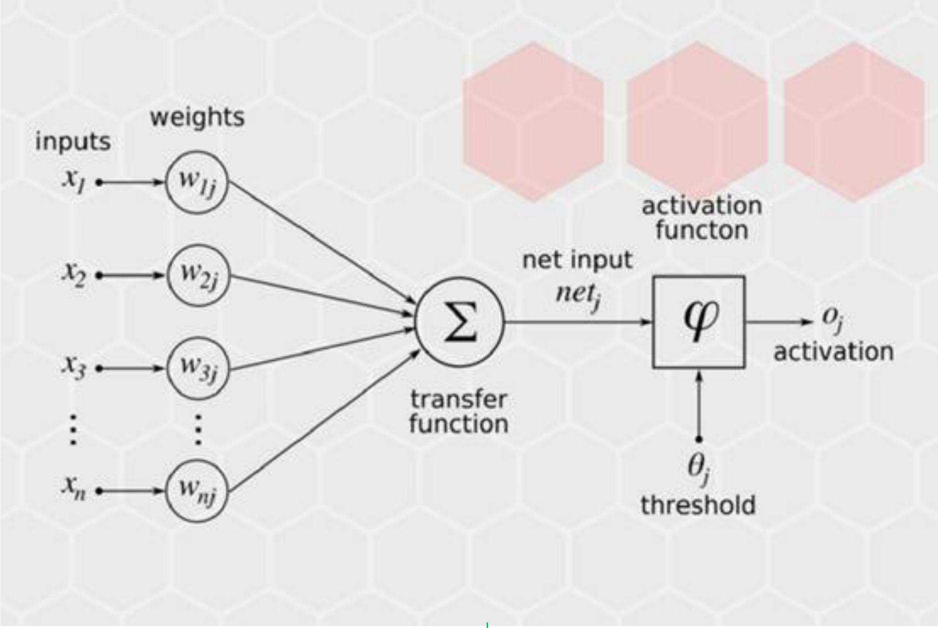
分别为 预测值 真实值

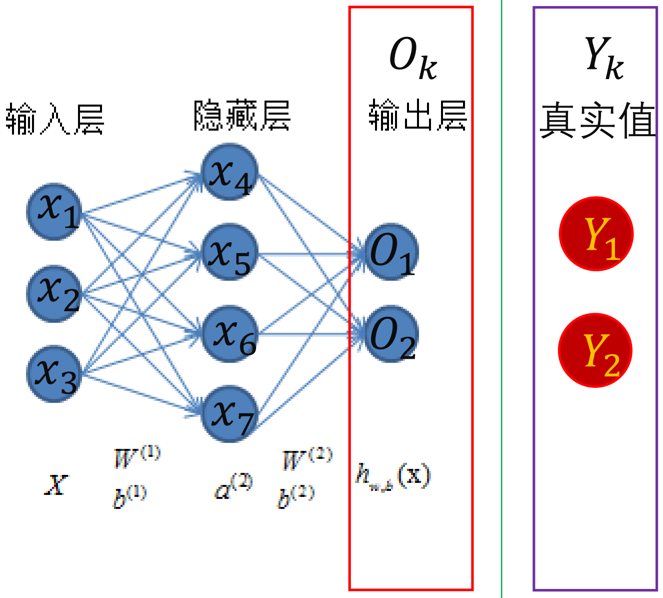
五、激活函数与损失函数的区别与联系

两者的用途不同、位置不同

除了输入层的神经元的值不需要激活外，其它层（即隐层和输出层）**每个神经元都需要激活函数**，常用的激活函数有Sigmoid，ReLU，Softmax等，作用是非线性变换，增加非线性因素。

损失函数仅作用于输出层的神经元，且为激活后的神经元的值上面，即作用于输出层激活后的神经元的值之上。损失函数即代价函数，是衡量/评估预测值和真实值之间的差距/误差的函数。





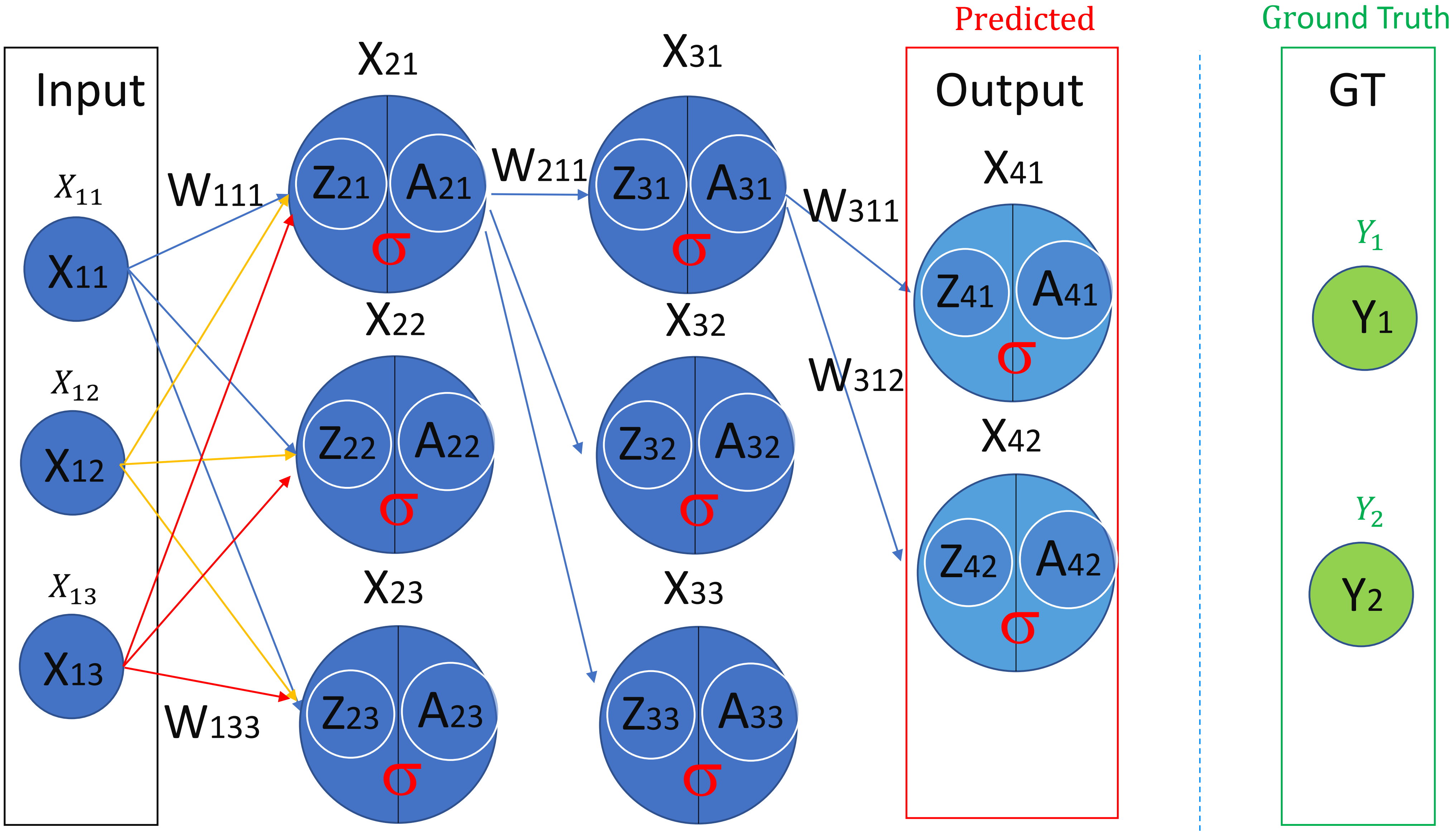
Part 03-2

Backpropagation

人工神经网络误差反向传播原理

误差反向传播，权值更新

一、一个简单的神经网络及其符号表示

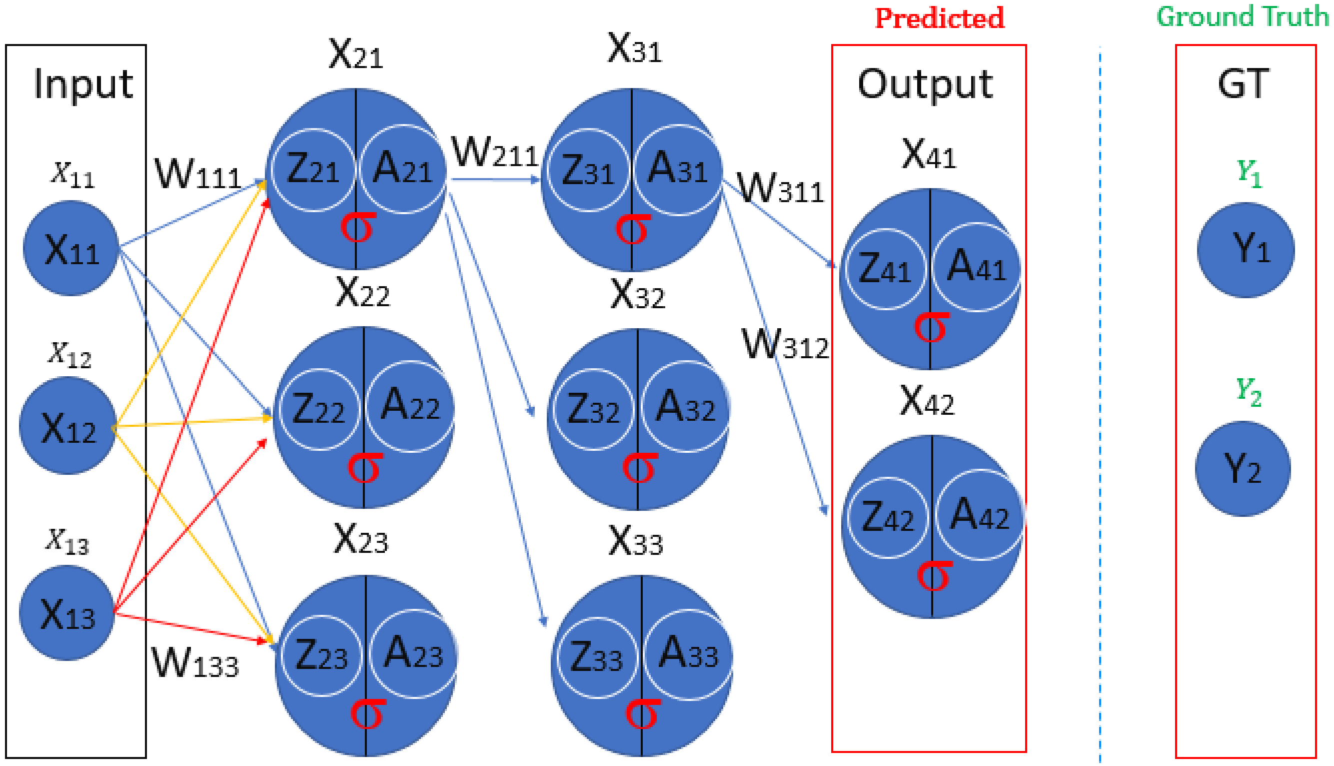


详见本书教材第41页

二、输出层神经元的求导公式（1）

$A_{41}=f(Z_{41})$

$Z_{41}=A_{31}*W_{311}+A_{32}*W_{321}+A_{33}*W_{331}+B_{41}$



均方差损失函数求导

$$\frac{d}{dO_k}L=\frac{d}{dO_k}\left(\frac{1}{2}\sum_{k=1}^{C}(Y_k-O_k)^2\right)$$

$$=O_k-Y_k$$

$$\frac{d}{dO_1}L=O_1-Y_1$$

$$\frac{d}{dO_2}L=O_2-Y_2$$

假定损失函数是均方差损失，激活函数均为Sigmoid

$$\frac{d}{dA_{41}}L=A_{41}-Y_1$$

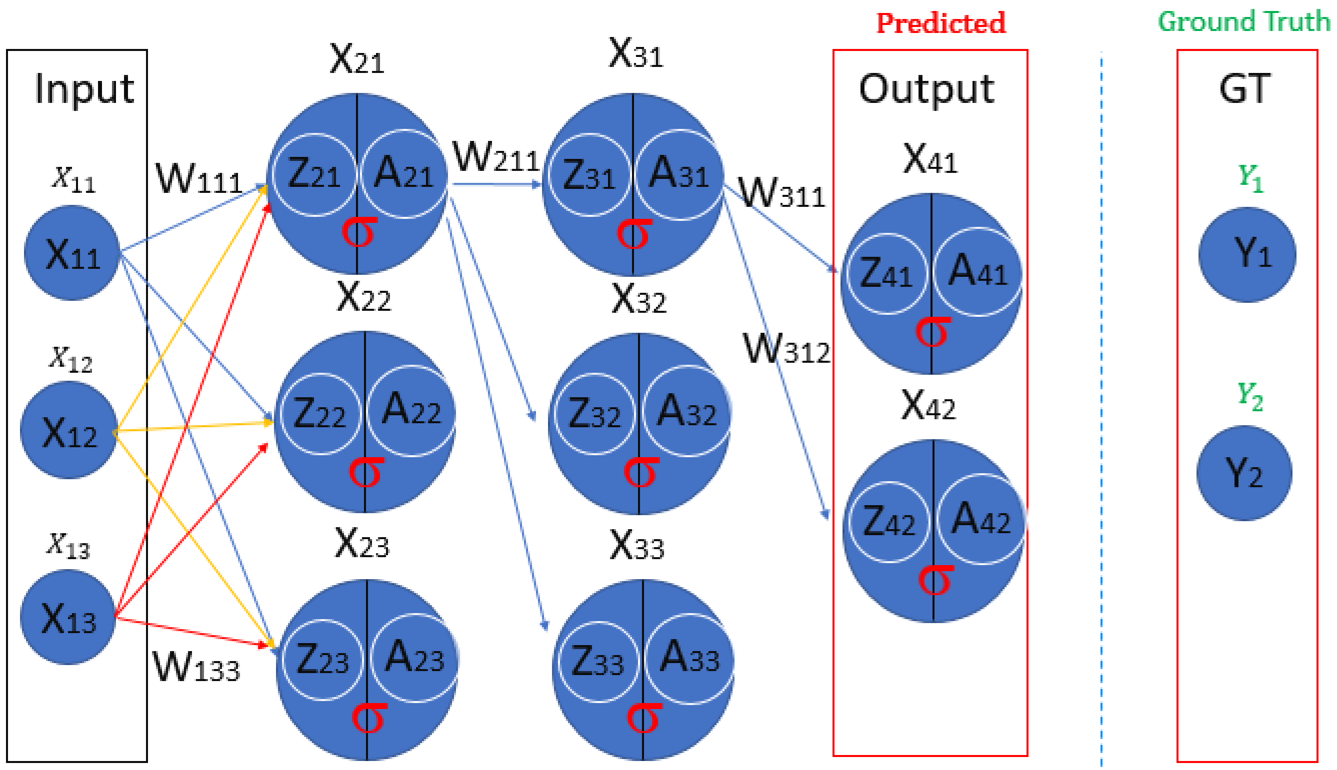
$$\frac{d}{dA_{42}}L=A_{42}-Y_2$$

详见本书教材第44页

二、输出层神经元的求导公式（2）

$A_{41}=f(Z_{41})$

$Z_{41}=A_{31}*W_{311}+A_{32}*W_{321}+A_{33}*W_{331}+B_{41}$



均方差损失函数求导

$$\frac{d}{dO_k}L=\frac{d}{dO_k}\left(\frac{1}{2}\sum_{k=1}^{C}(Y_k-O_k)^2\right)=O_k-Y_k$$

$$\frac{d}{dA_{41}}L=A_{41}-Y_1$$

$$\frac{d}{dA_{42}}L=A_{42}-Y_2$$

$$\frac{d}{dZ_{41}}L=\frac{d}{dA_{41}}L*\frac{d}{dZ_{41}}A_{41}$$

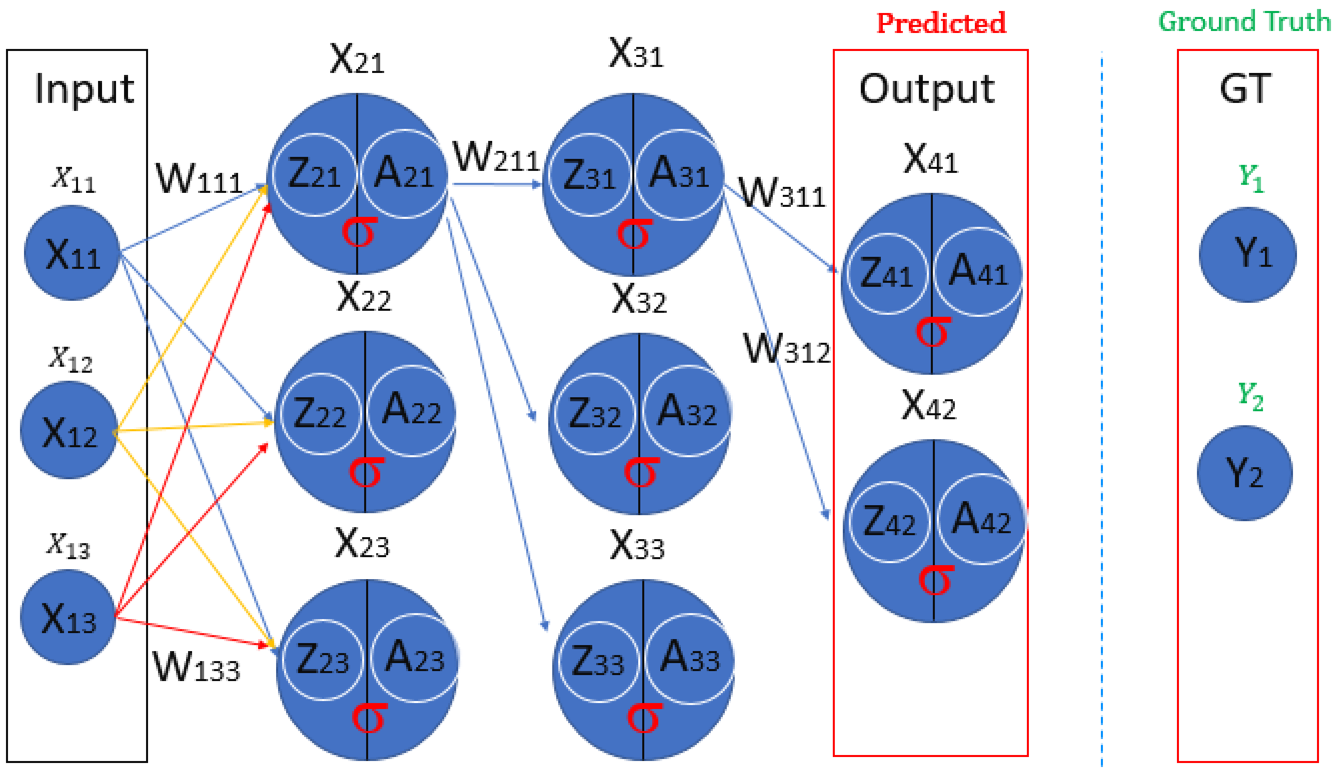
$$=(A_{41}-Y_1)*f(Z_{41})*(1-f(Z_{41}))=(A_{41}-Y_1)*A_{41}*(1-A_{41})$$

$$\frac{d}{dZ_{42}}L=(A_{42}-Y_2)*A_{42}*(1-A_{42})$$

二、输出层神经元的求导公式（3）

$A_{41}=f(Z_{41})$

$Z_{41}=A_{31}*W_{311}+A_{32}*W_{321}+A_{33}*W_{331}+B_{41}$



均方差损失函数求导

$$\frac{d}{dO_k}L=\frac{d}{dO_k}\left(\frac{1}{2}\sum_{k=1}^{C}(Y_k-O_k)^2\right)$$

$$=O_k-Y_k$$

$$\frac{d}{dA_{41}}L=A_{41}-Y_1$$

$$\frac{d}{dA_{42}}L=A_{42}-Y_2$$

$$\frac{d}{dZ_{41}}L=(A_{41}-Y_1)*A_{41}*(1\text{-}A_{41})$$

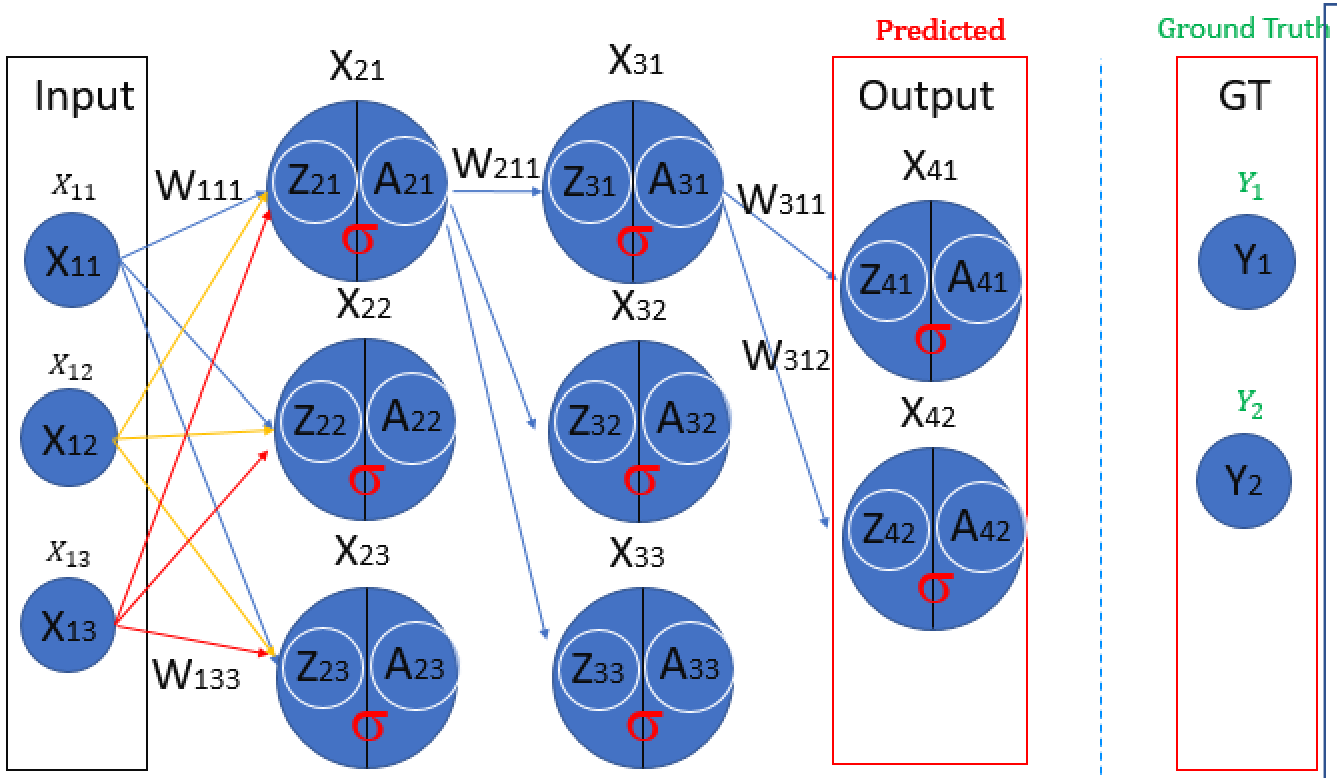
$$\frac{d}{dZ_{42}}L=(A_{42}-Y_2)*A_{42}*(1\text{-}A_{42})$$

假定输出层和隐层，都使用Sigmoid激活函数

三、隐层神经元的求导公式（1）

$A_{41}=f(Z_{41})$

$Z_{41}=A_{31}*W_{311}+A_{32}*W_{321}+A_{33}*W_{331}+B_{41}$



输出层神经元求导

$$\frac{d}{dO_k}L=\frac{d}{dO_k}\left(\frac{1}{2}\sum_{k=1}^{C}(Y_k-O_k)^2\right)$$

$$=O_k-Y_k$$

$$\frac{d}{dA_{41}}L=A_{41}-Y_1$$

$$\frac{d}{dA_{42}}L=A_{42}-Y_2$$

$$\frac{d}{dZ_{41}}L=(A_{41}-Y_1)*A_{41}*(1-A_{41})$$

δZ_{41}

$$\frac{d}{dZ_{42}}L=(A_{42}-Y_2)*A_{42}*(1-A_{42})$$

δZ_{42}

$$\frac{d}{dZ_{31}}L=\frac{d}{dZ_{41}}L*\frac{d}{dA_{31}}Z_{41}*\frac{d}{dZ_{31}}A_{31}+$$

隐层

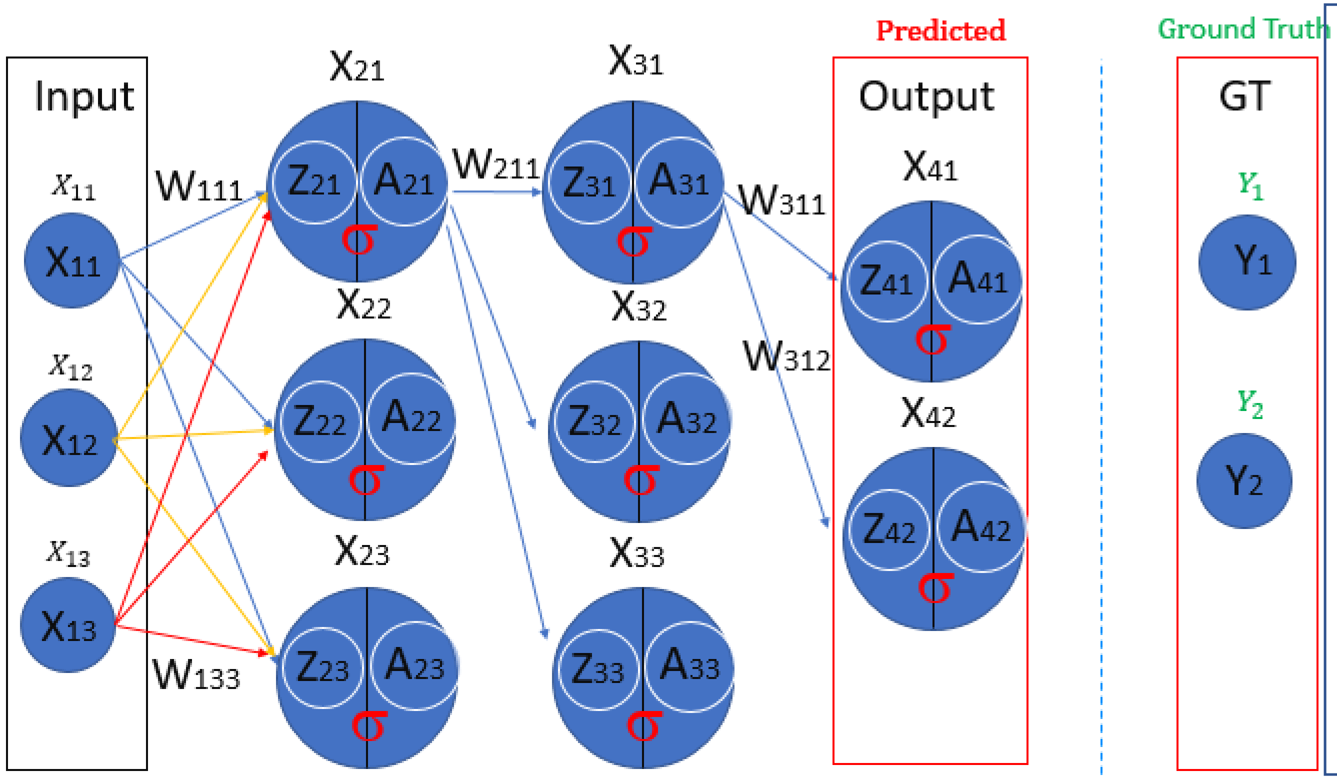
$$\frac{d}{dZ_{42}}L*\frac{d}{dA_{31}}Z_{42}*\frac{d}{dZ_{31}}A_{31}$$

详见本书教材第45页

三、隐层神经元的求导公式（2）

$A_{41}=f(Z_{41})$

$Z_{41}=A_{31}*W_{311}+A_{32}*W_{321}+A_{33}*W_{331}+B_{41}$



输出层神经元求导

$$\frac{d}{dO_k}L=\frac{d}{dO_k}\left(\frac{1}{2}\sum_{k=1}^{C}(Y_k-O_k)^2\right)$$

$$=O_k-Y_k$$

δZ_{41}

$$\frac{d}{dZ_{41}}L=(A_{41}-Y_1)*A_{41}*(1\text{-}A_{41})$$

δZ_{42}

$$\frac{d}{dZ_{42}}L=(A_{42}-Y_2)*A_{42}*(1\text{-}A_{42})$$

$$\frac{d}{dZ_{31}}L=\frac{d}{dZ_{41}}L*\frac{d}{dA_{31}}Z_{41}*\frac{d}{dZ_{31}}A_{31}+\frac{d}{dZ_{42}}L*\frac{d}{dA_{31}}Z_{42}*\frac{d}{dZ_{31}}A_{31}$$

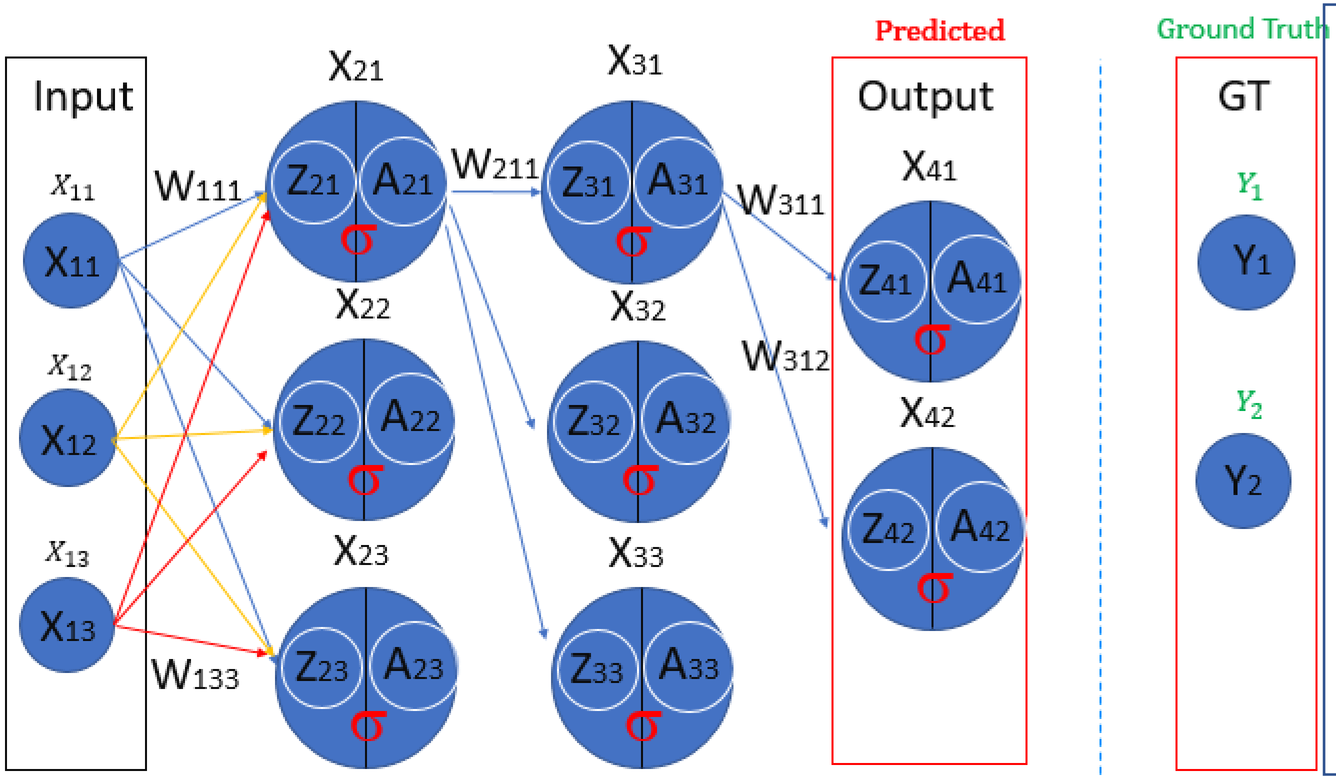
$$=\delta Z_{41}*W_{311}*\frac{d}{dZ_{31}}A_{31}+\delta Z_{42}*W_{312}*\frac{d}{dZ_{31}}A_{31}$$

$$=(\delta Z_{41}*W_{311}+\delta Z_{42}*W_{312})*A_{31}*(1\text{-}A_{31})$$

三、隐层神经元的求导公式（3）

$A_{41}=f(Z_{41})$

$Z_{41}=A_{31}*W_{311}+A_{32}*W_{321}+A_{33}*W_{331}+B_{41}$



输出层神经元求导

$$\frac{d}{dO_k}L=\frac{d}{dO_k}\left(\frac{1}{2}\sum_{k=1}^{C}(Y_k-O_k)^2\right)$$

$$=O_k-Y_k$$

$$\frac{d}{dZ_{41}}L=(A_{41}-Y_1)*\underbrace{A_{41}*(1\text{-}A_{41})}_{}\quad \delta Z_{41}$$

$$\frac{d}{dZ_{42}}L=(A_{42}-Y_2)*A_{42}*(1\text{-}A_{42})\quad \delta Z_{42}$$

$$\frac{d}{dZ_{31}}L=\frac{d}{dZ_{41}}L*\frac{d}{dA_{31}}Z_{41}*\frac{d}{dZ_{31}}A_{31}+\frac{d}{dZ_{42}}L*\frac{d}{dA_{31}}Z_{42}*\frac{d}{dZ_{31}}A_{31}$$

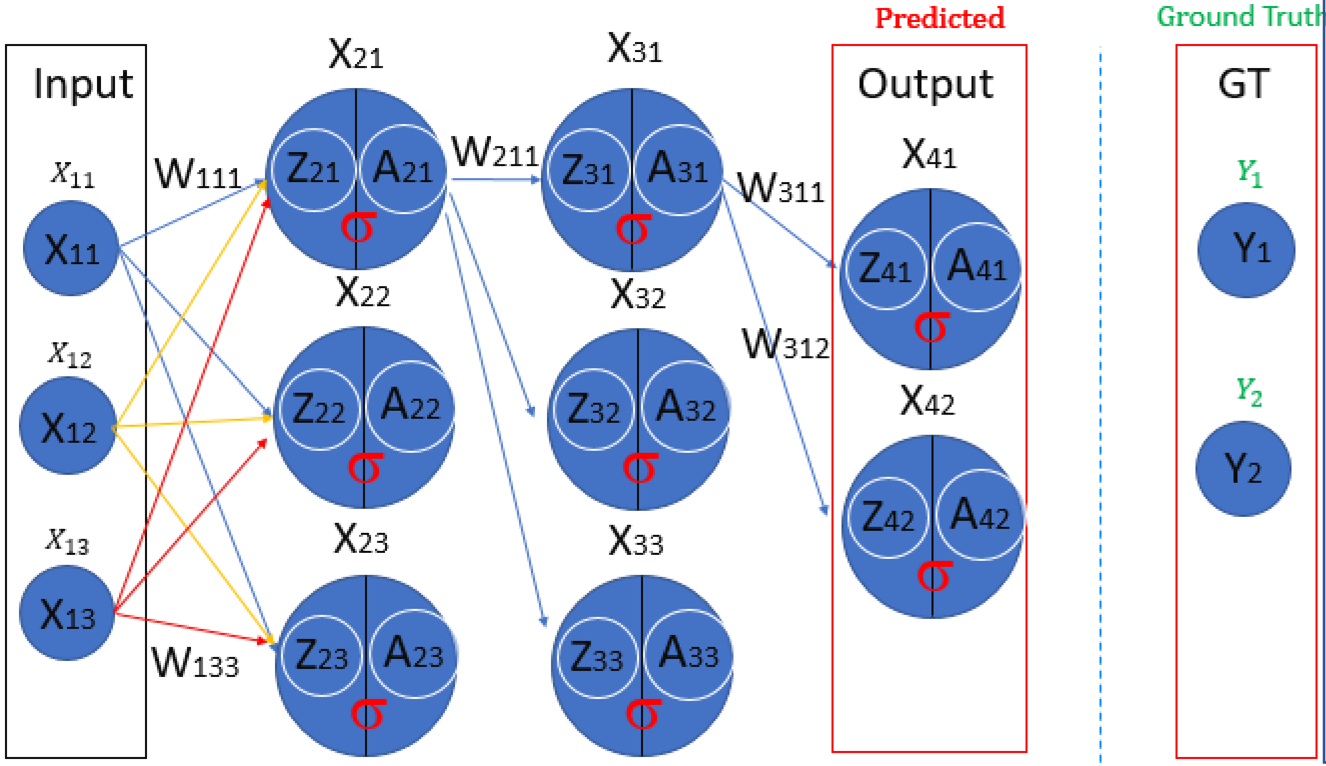
$$=(\delta Z_{41}*W_{311}+\delta Z_{42}*W_{312})*A_{31}*(1\text{-}A_{31})$$

$$=A_{31}*(1\text{-}A_{31})*\sum_{k=1}^{2}(\delta Z_{4k}*W_{31k})$$

三、隐层神经元的求导公式（4）

$A_{41}=f(Z_{41})$

$Z_{41}=A_{31}*W_{311}+A_{32}*W_{321}+A_{33}*W_{331}+B_{41}$



输出层神经元求导

$$\frac{d}{dO_k}L=\frac{d}{dO_k}\left(\frac{1}{2}\sum_{k=1}^{C}(Y_k-O_k)^2\right)$$

$$=O_k-Y_k$$

$$\frac{d}{dZ_{41}}L=(A_{41}-Y_1)*A_{41}*(1\text{-}A_{41}) \quad \delta Z_{41}$$

$$\frac{d}{dZ_{42}}L=(A_{42}-Y_2)*A_{42}*(1\text{-}A_{42}) \quad \delta Z_{42}$$

$$\frac{d}{dZ_{31}}L=A_{31}*(1\text{-}A_{31})*\sum_{k=1}^{2}(\delta Z_{4k}*W_{31k})$$

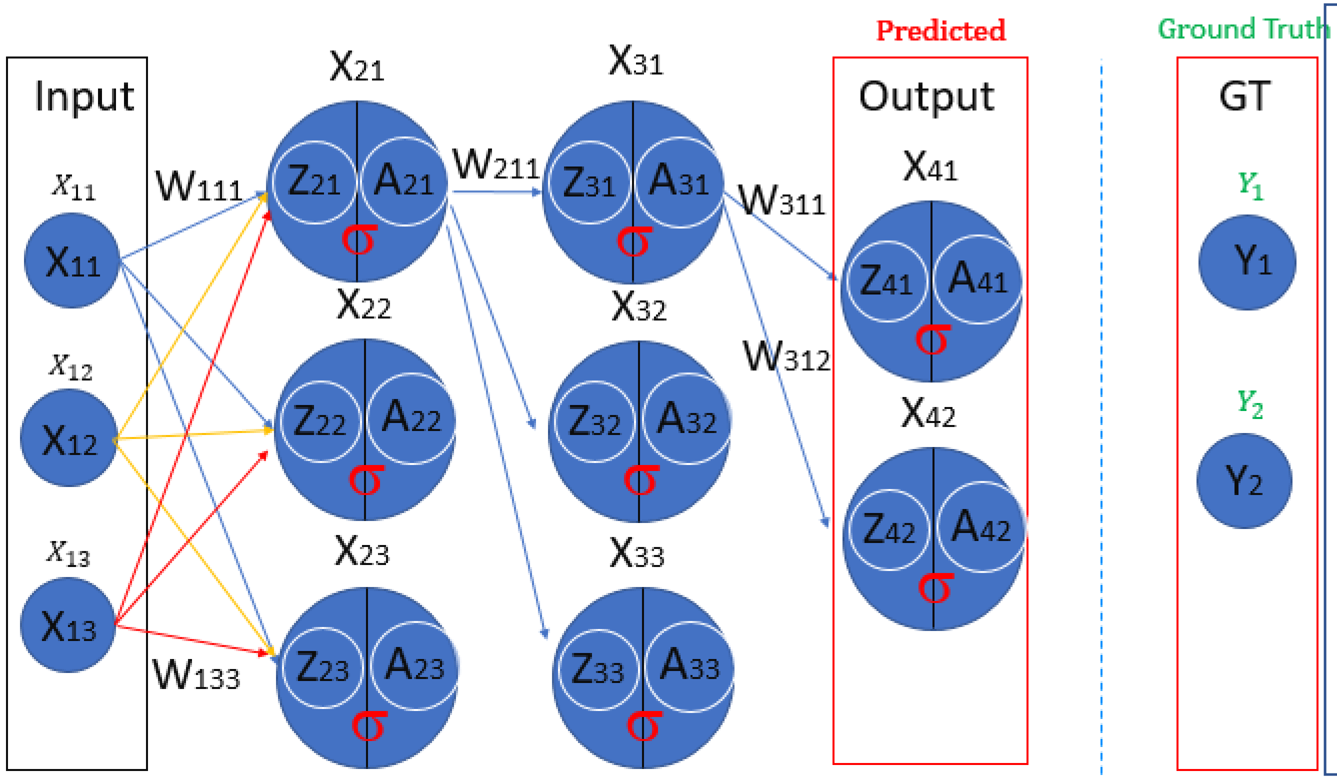
$$\frac{d}{dZ_{ij}}L=A_{ij}*(1\text{-}A_{ij})*\sum_{k=1}^{N_{i+1}}(\delta Z_{(i+1)k}*W_{ijk})$$

$$\delta Z_{ij}=A_{ij}*(1\text{-}A_{ij})*\sum_{k=1}^{N_{i+1}}(\delta Z_{(i+1)k}*W_{ijk})$$

三、隐层神经元的求导公式（5）

$A_{41}=f(Z_{41})$

$Z_{41}=A_{31}*W_{311}+A_{32}*W_{321}+A_{33}*W_{331}+B_{41}$



输出层神经元求导

$$\frac{d}{dO_k}L=\frac{d}{dO_k}\left(\frac{1}{2}\sum_{k=1}^{C}(Y_k-O_k)^2\right)$$

$$=O_k-Y_k$$

$$\frac{d}{dZ_{41}}L=(A_{41}-Y_1)*A_{41}*(1-A_{41})\quad \delta Z_{41}$$

$$\frac{d}{dZ_{42}}L=(A_{42}-Y_2)*A_{42}*(1-A_{42})\quad \delta Z_{42}$$

$$\delta Z_{ij}=A_{ij}*(1-A_{ij})*\sum_{k=1}^{N_{i+1}}(\delta Z_{(i+1)k}*W_{ijk})$$

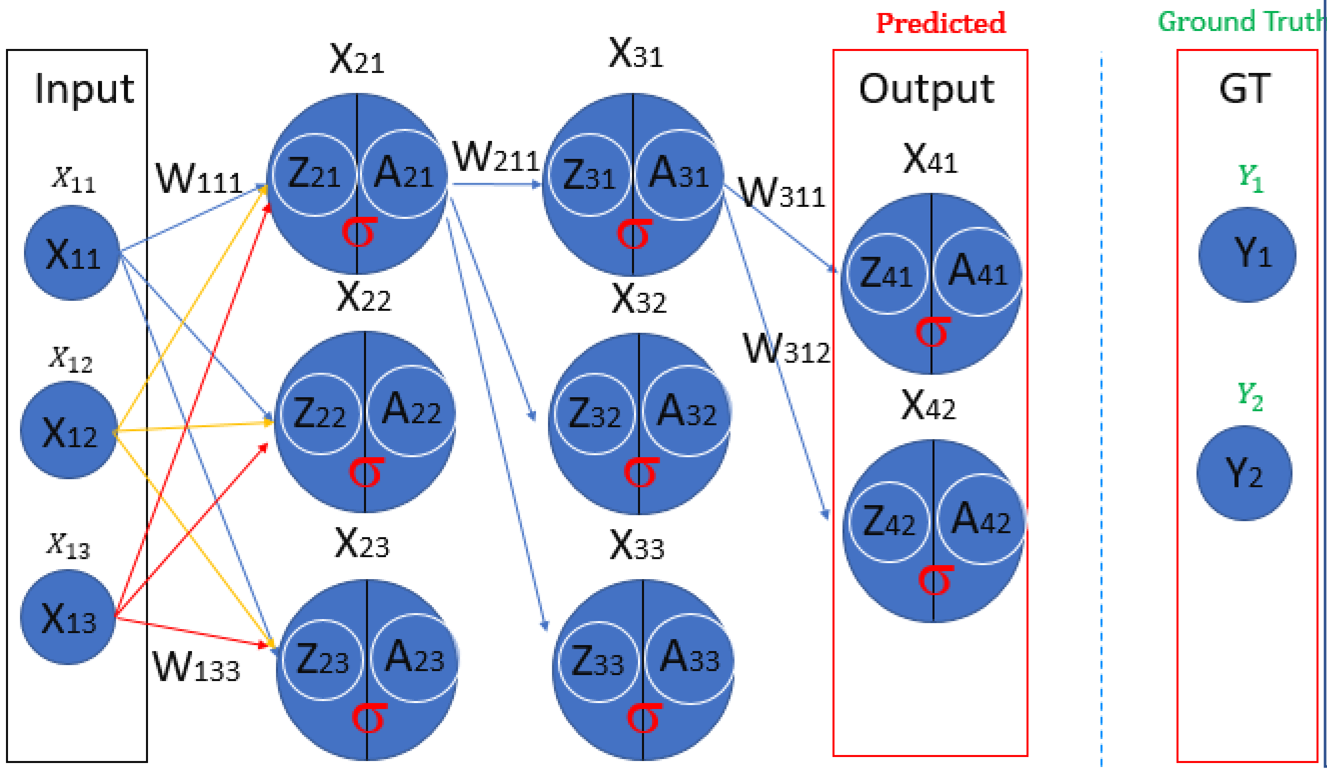
$$\frac{d}{dW_{311}}L=\frac{d}{dZ_{41}}L*\frac{d}{dW_{311}}Z_{41}=\delta Z_{41}*A_{31}$$

$$\frac{d}{dW_{ijk}}L=\boxed{A_{ij}*\delta Z_{(i+1)k}}$$

四、权值和偏置的求导公式（1）

$A_{41}=f(Z_{41})$

$Z_{41}=A_{31}*W_{311}+A_{32}*W_{321}+A_{33}*W_{331}+B_{41}$



输出层神经元求导

$$\frac{d}{dO_k}L=\frac{d}{dO_k}\left(\frac{1}{2}\sum_{k=1}^{C}(Y_k-O_k)^2\right)$$

$$=O_k-Y_k$$

$$\frac{d}{dZ_{41}}L=(A_{41}-Y_1)*\underbrace{A_{41}*(1\text{-}A_{41})}_{}\quad \delta Z_{41}$$

$$\frac{d}{dZ_{42}}L=(A_{42}-Y_2)*A_{42}*(1\text{-}A_{42})\quad \delta Z_{42}$$

$$\delta Z_{ij}=A_{ij}*(1-A_{ij})*\sum_{k=1}^{N_{i+1}}(\delta Z_{(i+1)k}*W_{ijk})$$

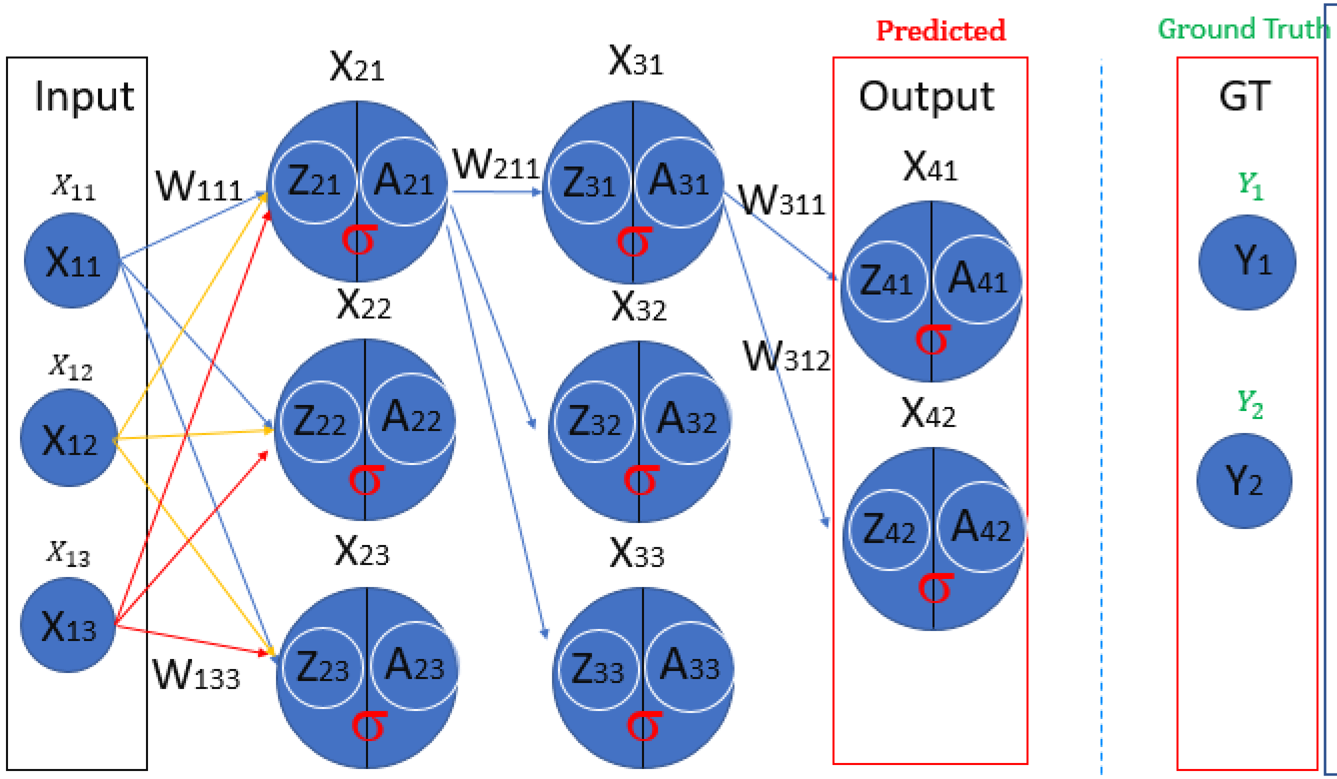
$$\frac{d}{dW_{ijk}}L=\delta W_{ijk}=A_{ij}*\delta Z_{(i+1)k}$$

$$\frac{d}{dB_{ik}}L=\frac{d}{dZ_{ik}}L*\frac{d}{dB_{ik}}Z_{ik}=\frac{d}{dZ_{ik}}L=\delta Z_{ik}$$

四、权值和偏置的求导公式（2）

$A_{41}=f(Z_{41})$

$Z_{41}=A_{31}{}^{*}W_{311}+A_{32}{}^{*}W_{321}+A_{33}{}^{*}W_{331}+B_{41}$



输出层神经元求导

$$\frac{d}{dO_k}L=\frac{d}{dO_k}\left(\frac{1}{2}\sum_{k=1}^{C}(Y_k-O_k)^2\right)$$

$$=O_k-Y_k$$

δZ_{41}

$$\frac{d}{dZ_{41}}L=(A_{41}-Y_1)*A_{41}*(1\text{-}A_{41})$$

δZ_{42}

$$\frac{d}{dZ_{42}}L=(A_{42}-Y_2)*A_{42}*(1\text{-}A_{42})$$

$$\delta Z_{ij}=A_{ij}*(1-A_{ij})*\sum_{k=1}^{N_{i+1}}(\delta Z_{(i+1)k}*W_{ijk})$$

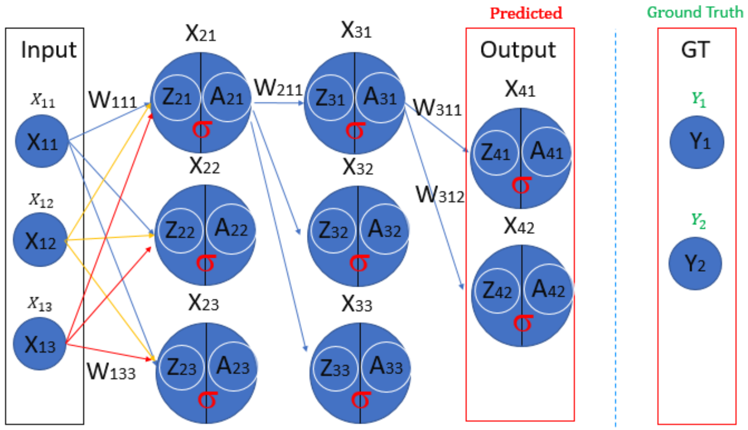
$$\delta W_{ijk}=A_{ij}*\delta Z_{(i+1)k}$$

$$\delta B_{ij}=\delta Z_{ij}$$

五、人工神经网络求导总结

$$A_{41}=f(Z_{41})$$

$$Z_{41}=A_{31}*W_{311}+A_{32}*W_{321}+A_{33}*W_{331}+B_{41}$$



符号说明

L表示的总损失公式，默认用均方差损失

n 是 神经网络的总层数（含输入输出层）

N_i是神经网络的第i层的神经元的数量

W_{ijk}是从神经网络的第i层第j个神经元到第(i+1)层第k个神经元的权重的值

B_{ik}是神经网络的第i层第k个神经元所对应的偏移量的值

A_{ij}是神经网络第i层第j个神经元激活后的值

Z_{ij}是神经网络第i层第j个神经元激活前的值

δZ_{ij}是神经网络的第i层的第j个神经元在激活前，总损失L与之的偏导数

δB_{ik}是总损失L与B_{ik}的偏导数

δW_{ijk}是总损失L与W_{ijk}的偏导数

1. 输出层神经元（激活前）求导

\# 当i=n(输出层)，n为总层数; j=1...N_i

$$\delta Z_{ij} = (A_{ij} - Y_j) * A_{ij} * (1-A_{ij}) \quad 或$$

$$\delta Z_{nj} = (A_{nj} - Y_j) * A_{nj} * (1-A_{nj})$$

2. 隐层神经元（激活前）求导

\# 当i=n-1，n-2, ..., 2; j=1...N_i

$$\delta Z_{ij} = A_{ij} * (1- A_{ij}) * \sum_{k=1}^{N_{i+1}} (\delta Z_{(i+1)k} * W_{ijk})$$

$$\delta W_{ijk} = A_{ij} * \delta Z_{(i+1)k}$$

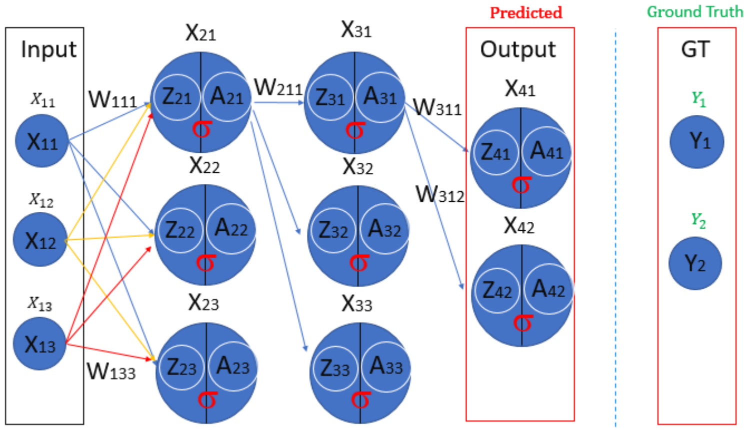
$$\delta B_{ij} = \delta Z_{ij}$$

六、人工神经网络求导公式汇总

$A_{41}=f(Z_{41})$

$Z_{41}=A_{31}*W_{311}+A_{32}*W_{321}+A_{33}*W_{331}+B_{41}$

#1. **输出层**单元j误差项的计算 δZ_{nj}



$$\delta O_1=(O_1-Y_1)*O_1(1-O_1) \qquad \delta Z_{n1}$$

$$\delta O_2=(O_2-Y_2)*O_2(1-O_2) \qquad \delta Z_{n2}$$

#2. **隐含层**单元j误差项的计算， k为下一层与隐藏单元j相连的单元

j=n-1…2

$$\delta Z_{ij}=A_{ij}*(1-A_{ij})*\underbrace{\sum_{k=1}^{N_{i+1}}}_{\text{下一层}}(\delta Z_{(i+1)k}*W_{ijk})$$

$$\delta B_{ij}=\delta Z_{ij}$$

#3. 隐含层单元i到下一层单元j之间的**权重W_{ij}**的误差项的计算，Oi是第i层的输出值， δ_j是j隐藏单元的误差项

$$\delta W_{ijk}=A_{ij}*\delta Z_{(i+1)k}$$

最后注意： δZ_{ij}表示的是激活前的某个神经元的偏导值；而Z_{ij}激活后是A_{ij}

七、人工神经网络前向传播与误差反向传播算法实现

1. 前向传播算法

#1. 首先输入神经网络的总层数及每层的
  神经元的数量
n=4
N = [4, 3, 3, 3, 2]

#2.初始化 二维数组A[][]， Z[][]， B[][];
  初始化三维数组W[][][];三维数组DW[][][]
  初始化 二维数组DZ[][]， DB[][];
  输入数据作为A[][]的第一层的各神经元的值；
  其期望输出(Ground Truth)作为参照/目标值。

#3. 从第2层开始到第n层，依次计算Z_{ij} A_{ij}
For i=2 … n
  For k= 1 … N_i
    For j=1 … N_{i-1}
      Z_{ik} + = $A_{(i-1)j}$ * $W_{(i-1)jk}$
    Z_{ik} + = B_{ik}
    A_{ik} = f(Z_{ik})

2. BP误差反向传播算法

#1. 首先计算输出层(第n层)的各偏导值δZ_{nj}
For j=1… N_n
  $DZ_{nj} = (A_{nj} - Y_j) * A_{nj} * (1-A_{nj})$

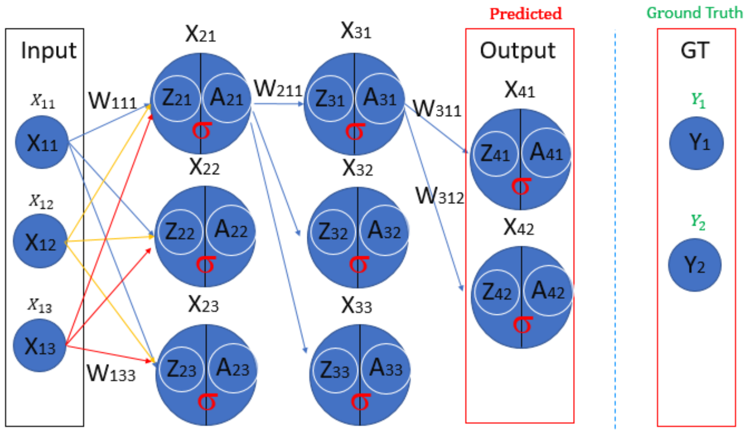
#2. 然后从第n-1层开始，依次计算δZ_{ij}
For i=n-1 … 2
  For j=1 … N_i
    $DZ_{ij} = A_{ij} * (1- A_{ij}) * \sum_{k=1}^{N_{i+1}} (\delta Z_{(i+1)k} * W_{ijk})$

#3. 从第n层(输出层)到第2层(即输入层之外的所有层)，计算δB_{ij}
For i=n … 2
  For j=1 … N_i
    DB_{ij} = DZ_{ij}
    B_{ij} - = λ* DB_{ij} #更新偏移量的值， λ为学习率=0.1

#4. 从第n-1层(第一个隐层)到第1层，计算δW_{ijk} ； 更新权重值W_{ijk}
For i=n-1 … 1
  For j=1… N_i
    For k=1 … N_{i+1}
      DW_{ijk} = $A_{ij} * \delta Z_{(i+1)k}$ # 计算δW_{ijk}
      W_{ijk} - = λ* DW_{ijk} #更新权重的值， λ为学习率=0.1

详见本书教材第47-49页

练习题



w_{111}	0.5	w_{211}	0.5
w_{112}	0.2	w_{212}	-0.1
w_{113}	-0.5	w_{213}	0.6
w_{121}	-0.2	w_{221}	-0.2
w_{122}	0.3	w_{222}	0.1
w_{123}	0.4	w_{223}	0.2
w_{131}	0.8	w_{231}	-0.5
w_{132}	-0.6	w_{232}	0.4
w_{133}	0.4	w_{233}	0.6
w_{311}	0.3	b_{21}	0.2
w_{312}	-0.2	b_{22}	-0.5
w_{331}	0.2	b_{23}	0.4
w_{321}	0.5	b_{31}	0.6
w_{322}	0.2	b_{32}	0.6
w_{332}	0.4	b_{33}	0.6
		b_{41}	0.8
X11	1	b_{42}	0.2
X12	0	Y_1	0.9
X13	1	Y_2	0.1