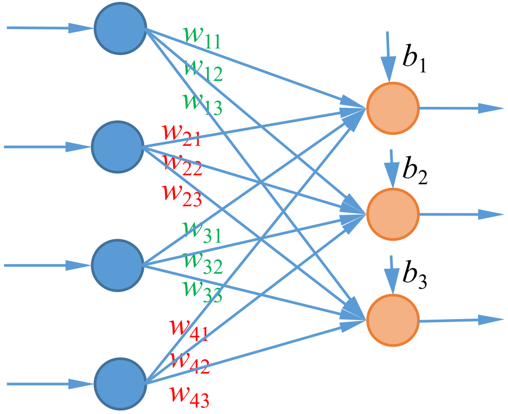
神经网络&深度学习 TENSORFLOW Google

12.4 误差反向传播算法

中国大学MOOC

感知机 / 单层神经网络：线性分类

多层神经网络：非线性分类

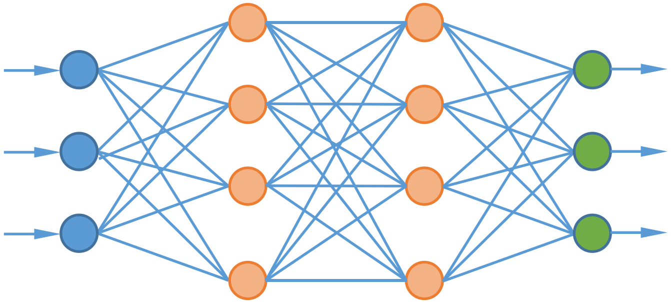


$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{31} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{bmatrix}$$

$$B = [\, b_1, b_2, b_3 \,]$$

$$W^{(k+1)} = W^{(k)} - \eta \frac{\partial Loss(W,B)}{\partial W}$$

$$B^{(k+1)} = B^{(k)} - \eta \frac{\partial Loss(W,B)}{\partial B}$$



□ 误差反向传播算法（Backpropagation，BP）

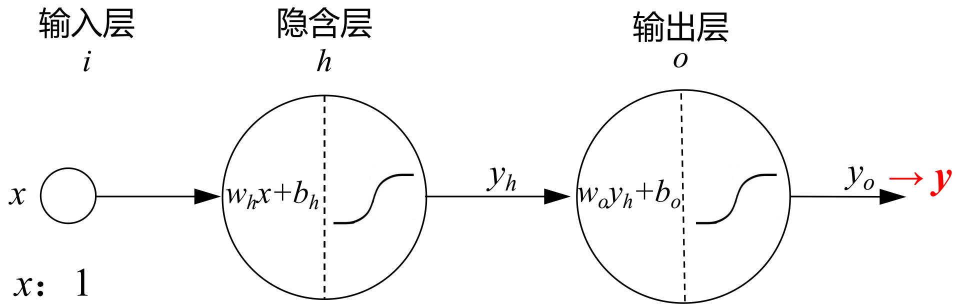
利用**链式法则**，**反向传播**损失函数的**梯度信息**，计算出损失函数对网络中所有模型参数的**梯度**

□ 神经网络的训练

- 使用**误差反向传播算法**计算梯度
- 使用**梯度下降法**学习模型参数

12.4 误差反向传播算法

1-1-1神经网络的误差反向传播



x：1

y：0.8

$$z_h = w_h x + b_h$$

$$y_h = \frac{1}{1+e^{-z_h}}$$

$$z_o = w_o y_h + b_o$$

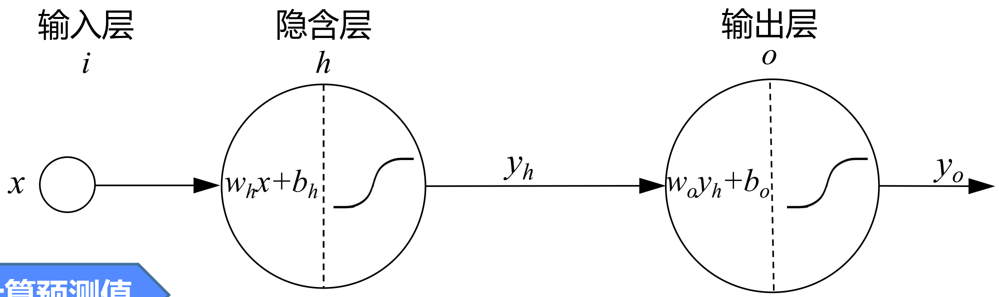
$$y_o = \frac{1}{1+e^{-z_o}}$$

12.4 误差反向传播算法

Step1: 设置模型参数初始值

$w_h=0.2,\ b_h=0.1,\ w_o=0.3,\ b_o=0.2$

x: 1
y: 0.8



Step2: 正向计算预测值

$$y_h=\frac{1}{1+e^{-(0.2\times1+0.1)}}=0.57 \qquad y_o=\frac{1}{1+e^{-(0.3\times0.57+0.2)}}=0.59$$

Step3: 计算误差

$$Loss=\frac{1}{2}(y-y_o)^2=0.02205$$

Step4: 误差反向传播

$$w_o^{(k+1)}=w_o^{(k)}-\eta\frac{\partial Loss}{\partial w_o}$$

$$b_o^{(k+1)}=b_o^{(k)}-\eta\frac{\partial Loss}{\partial b_o}$$

12.4 误差反向传播算法

链式求导法则

$$\frac{\partial Loss}{\partial w_o} = \frac{\partial Loss}{\partial y_o} \cdot \frac{\partial y_o}{\partial z_o} \cdot \frac{\partial z_o}{\partial w_o} = 0.21 \times 0.2419 \times 0.57 = 0.02895543$$

$$\boxed{\eta = 0.5}$$

$$w_o^{(1)} = w_o^{(0)} - \eta \frac{\partial Loss}{\partial w_o}$$
$$= 0.3 - 0.5 \times 0.02895543$$
$$= 0.28552228$$

$$Loss = \frac{1}{2}(y - y_o)^2 \qquad \frac{\partial Loss}{\partial y_o} = 2 \times \frac{1}{2} \times (y - y_o) = 0.8 - 0.59 = \boxed{0.21}$$

$$y_o = \frac{1}{1 + e^{-z_o}} \qquad \frac{\partial y_o}{\partial z_o} = \frac{e^{-z_o}}{(1 + e^{-z_o})^2} = y_o(1 - y_o)$$
$$= 0.59 \times (1 - 0.59) = \boxed{0.2419}$$

$$z_o = w_o y_h + b_o \qquad \frac{\partial z_o}{\partial w_o} = y_h = \boxed{0.57}$$

链式求导法则

$\eta = 0.5$

$$\frac{\partial Loss}{\partial w_o} = \frac{\partial Loss}{\partial y_o} \cdot \frac{\partial y_o}{\partial z_o} \cdot \frac{\partial z_o}{\partial w_o} = 0.21 \times 0.2419 \times 0.57 = 0.02895543$$

$$w_o^{(1)} = w_o^{(0)} - \eta \frac{\partial Loss}{\partial w_o} = 0.28552228$$

$$\frac{\partial Loss}{\partial b_o} = \frac{\partial Loss}{\partial y_o} \cdot \frac{\partial y_o}{\partial z_o} \cdot \boxed{\frac{\partial z_o}{\partial b_o}} = 0.21 \times 0.2419 = 0.050799$$

$$b_o^{(1)} = b_o^{(0)} - \eta \frac{\partial Loss}{\partial b_o} = 0.1746005$$

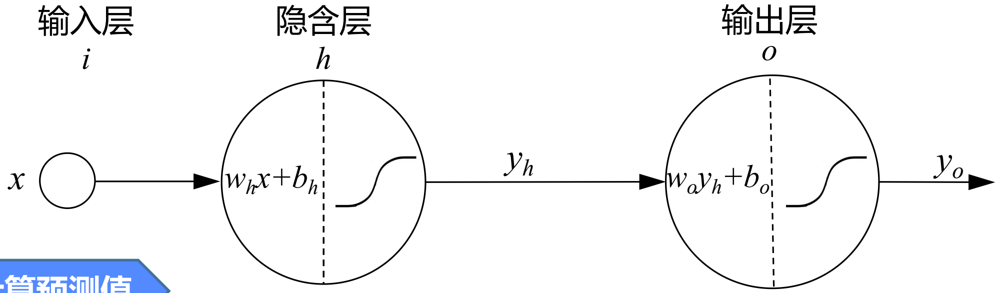
$$z_o = w_o y_h + b_o \qquad \frac{\partial z_o}{\partial b_o} = 1$$

12.4 误差反向传播算法

Step1: 设置模型参数初始值

w_h=0.2, b_h=0.1, w_o=0.3, b_o=0.2

x: 1
y: 0.8



Step2: 正向计算预测值

$$y_h=\frac{1}{1+e^{-(0.2\times1+0.1)}}=0.57 \qquad y_o=\frac{1}{1+e^{-(0.3\times0.57+0.2)}}=0.59$$

Step3: 计算误差

$$Loss=\frac{1}{2}(y-y_o)^2=0.02205$$

Step4: 误差反向传播

$$w_o^{(1)}=w_o^{(0)}-\eta\frac{\partial Loss}{\partial w_o}=0.02895543$$

$$b_o^{(1)}=b_o^{(0)}-\eta\frac{\partial Loss}{\partial b_o}=0.1746005$$

$$w_h^{(k+1)}=w_h^{(k)}-\eta\frac{\partial Loss}{\partial w_h}$$

$$b_h^{(k+1)}=b_h^{(k)}-\eta\frac{\partial Loss}{\partial b_h}$$

链式求导法则

$$\frac{\partial Loss}{\partial w_h}=\frac{\partial Loss}{\partial y_o}\cdot\frac{\partial y_o}{\partial z_o}\cdot\frac{\partial z_o}{\partial y_h}\cdot\frac{\partial y_h}{\partial z_h}\cdot\frac{\partial z_h}{\partial w_h}=0.21\times0.2419\times0.3\times0.2451\times1=0.00373525$$

$$Loss=\frac{1}{2}(y-y_o)^2 \qquad \frac{\partial Loss}{\partial y_o}=2\times\frac{1}{2}\times(y-y_o)=0.8-0.59=\boxed{0.21}$$

$$y_o=\frac{1}{1+e^{-z_o}} \qquad \frac{\partial y_o}{\partial z_o}=\frac{e^{-z_o}}{(1+e^{-z_o})^2}=y_o(1-y_o)=\boxed{0.2419}$$

$$z_o=w_o y_h+b_o \qquad \frac{\partial z_o}{\partial y_h}=w_o=\boxed{0.3}$$

$$y_h=\frac{1}{1+e^{-z_h}} \qquad \frac{\partial y_h}{\partial z_h}=y_h(1-y_h)=0.57\times(1-0.57)=\boxed{0.2451}$$

$$z_h=w_h x+b_h \qquad \frac{\partial z_h}{\partial w_h}=x=\boxed{1}$$

12.4 误差反向传播算法

链式求导法则

$$\frac{\partial Loss}{\partial b_h}=\frac{\partial Loss}{\partial y_o}\cdot\frac{\partial y_o}{\partial z_o}\cdot\frac{\partial z_o}{\partial y_h}\cdot\frac{\partial y_h}{\partial z_h}\cdot\boxed{\frac{\partial z_h}{\partial b_h}}=0.21\times0.2419\times0.3\times0.2451\times1=0.00373525$$

$$Loss=\frac{1}{2}(y-y_o)^2 \qquad \frac{\partial Loss}{\partial y_o}=2\times\frac{1}{2}\times(y-y_o)=0.8-0.59=0.21$$

$$y_o=\frac{1}{1+e^{-z_o}} \qquad \frac{\partial y_o}{\partial z_o}=\frac{e^{-z_o}}{(1+e^{-z_o})^2}=y_o(1-y_o)=0.2419$$

$$z_o=w_o y_h+b_o \qquad \frac{\partial z_o}{\partial y_h}=w_o=0.3$$

$$y_h=\frac{1}{1+e^{-z_h}} \qquad \frac{\partial y_h}{\partial z_h}=y_h(1-y_h)=0.57\times(1-0.57)=0.2451$$

$$z_h=w_h x+b_h \qquad \boxed{\frac{\partial z_h}{\partial b_h}=1}$$

链式求导法则

$$\frac{\partial Loss}{\partial w_h}=\frac{\partial Loss}{\partial y_o}\cdot\frac{\partial y_o}{\partial z_o}\cdot\frac{\partial z_o}{\partial y_h}\cdot\frac{\partial y_h}{\partial z_h}\cdot\frac{\partial z_h}{\partial w_h}=0.21\times0.2419\times0.3\times0.2451\times1=0.00373525$$

$$\frac{\partial Loss}{\partial b_h}=\frac{\partial Loss}{\partial y_o}\cdot\frac{\partial y_o}{\partial z_o}\cdot\frac{\partial z_o}{\partial y_h}\cdot\frac{\partial y_h}{\partial z_h}\cdot\frac{\partial z_h}{\partial b_h}=0.21\times0.2419\times0.3\times0.2451\times1=0.00373525$$

迭代公式

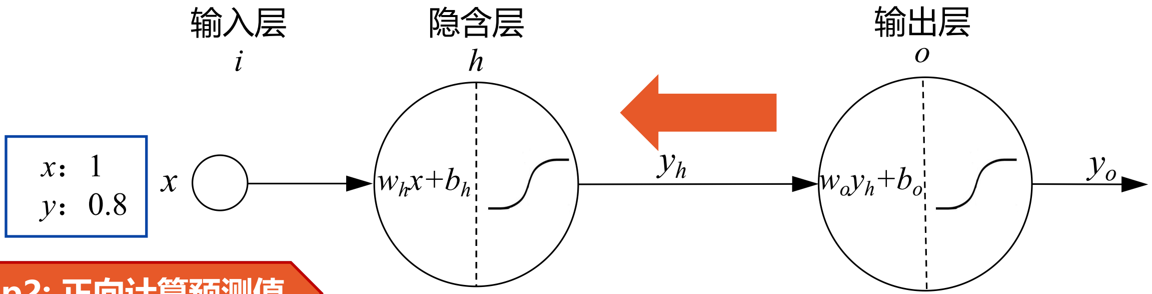
$$w_h^{(1)}=w_h^{(0)}-\eta\frac{\partial Loss}{\partial w_h}=0.2-0.5\times0.00373525=0.19813238$$

$$b_h^{(1)}=b_h^{(0)}-\eta\frac{\partial Loss}{\partial b_h}=0.1-0.5\times0.00373525=0.09813238$$

12.4 误差反向传播算法

Step1: 设置模型参数初始值

w_h=0.2，b_h=0.1，w_o=0.3，b_o=0.2



Step2: 正向计算预测值

$$y_h=\frac{1}{1+e^{-(0.2\times1+0.1)}}=0.57 \qquad y_o=\frac{1}{1+e^{-(0.3\times0.57+0.2)}}=0.59$$

Step3: 计算误差

$$Loss=\frac{1}{2}(y-y_o)^2=0.02205$$

$$w_h^{(1)}=w_h^{(0)}-\eta\frac{\partial Loss}{\partial w_h}=0.19813238$$

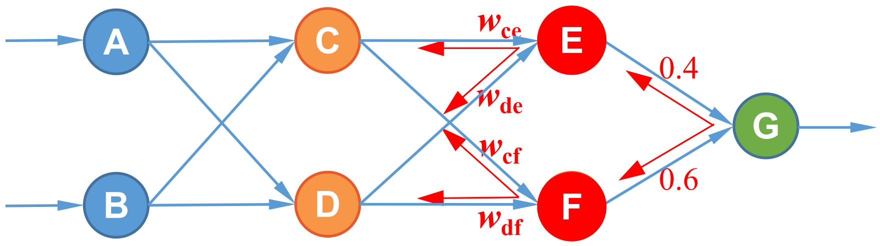
$$b_h^{(1)}=b_h^{(0)}-\eta\frac{\partial Loss}{\partial b_h}=0.09813238$$

$$w_o^{(1)}=w_o^{(0)}-\eta\frac{\partial Loss}{\partial w_o}=0.02895543$$

$$b_o^{(1)}=b_o^{(0)}-\eta\frac{\partial Loss}{\partial b_o}=0.1746005$$

Step4: 误差反向传播

隐含层有多个神经元的误差反向传播

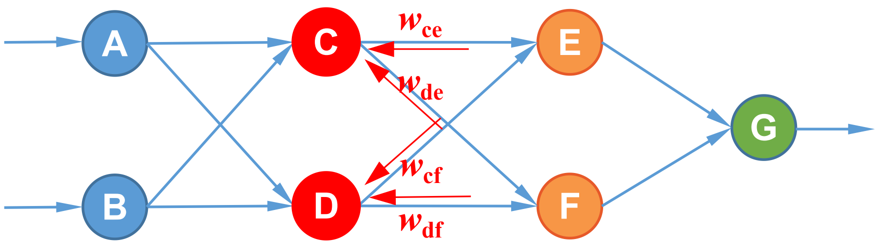


$$LossE = 0.4LossG$$
$$LossF = 0.6LossG$$

$$LossC_E = \frac{w_{ce}}{w_{ce}+w_{de}}LossE$$
$$LossD_E = \frac{w_{de}}{w_{ce}+w_{de}}LossE$$

$$LossC_F = \frac{w_{cf}}{w_{cf}+w_{df}}LossF$$
$$LossD_F = \frac{w_{df}}{w_{cf}+w_{df}}LossF$$

□ 隐含层有多个神经元的误差反向传播



$$LossE = 0.4LossG$$
$$LossF = 0.6LossG$$

$$LossC_E = \frac{w_{ce}}{w_{ce} + w_{de}} LossE$$

$$LossD_E = \frac{w_{de}}{w_{ce} + w_{de}} LossE$$

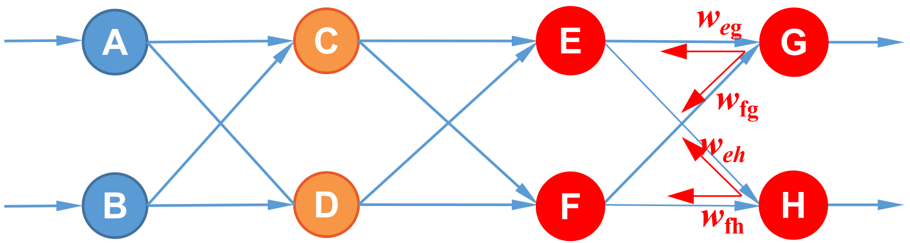
$$LossC = LossC_E + LossC_F = \frac{w_{ce}}{w_{ce} + w_{de}} LossE + \frac{w_{cf}}{wcf + wdf} LossF$$

$$LossD = LossD_E + LossD_F = \frac{w_{de}}{w_{ce} + w_{de}} LossE + \frac{w_{df}}{w_{cf} + w_{df}} LossF$$

$$LossC_F = \frac{w_{cf}}{w_{cf} + w_{df}} LossF$$

$$LossD_F = \frac{w_{df}}{w_{cf} + w_{df}} LossF$$

隐含层有多个神经元的误差反向传播



$$LossE=\frac{w_{eg}}{w_{eg}+w_{fg}}LossG+\frac{w_{eh}}{w_{eh}+w_{fh}}LossH$$

$$LossF=\frac{w_{fg}}{w_{eg}+w_{fg}}LossG+\frac{w_{fh}}{w_{eh}+w_{fh}}LossH$$

□ 多层神经网络的训练

通过**梯度下降法**训练模型参数
通过**误差反向传播法**计算梯度