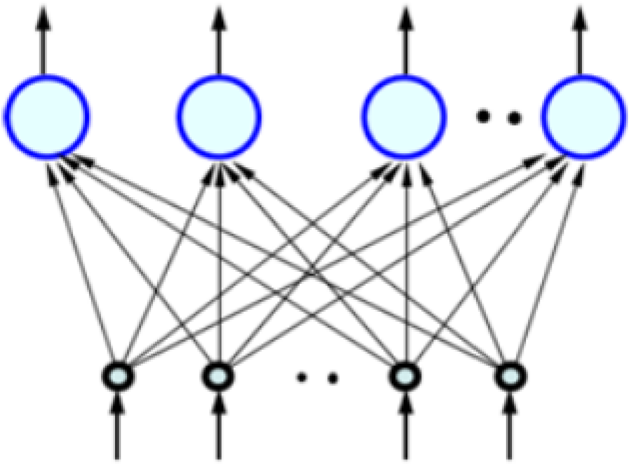
前反馈神经网络

一、在单层神经网络中，采用非线性单元时梯度下降法的公式推导。

单层神经网络描述：



输入层：d个输入节点，1个偏置节点

输入向量：$x = [1, x_1, x_2, \ldots, x_d]^T$

输出层：c个输出节点

输出向量：$z = [z_1, z_2, \ldots, z_c]^T$

权重集合：w_{ij}

对于输出层的节点j（c个输出），其输入加权和为：

$$sum_j = \sum_{i=0}^{d} w_{ij} x_i = W_j^T X,\ x_0 = 1\ (\ j = 1, 2 \ldots c)$$

$$W_j = [w_{0j}\ w_{1j}\ w_{2j}\ \ldots w_{dj}]^{\ T}$$

经过激励后的输出值为：

$$Z_j = f(sum_j)$$

给定训练样本$\{x_k\}_{k=1}^{n}$及相应的目标值$\{t_k\}_{k=1}^{n}$确定所有的连接权重w_{ij}

当采用非线性激励函数时：

损失函数：

$$Cost=\frac{1}{2}\sum_{k=1}^{n}\sum_{j=1}^{c}(t_j^k-z_j^k)^2=\frac{1}{2}\sum_{k=1}^{n}\sum_{j=1}^{c}(t_j^k-f(sum_j^k))^2,\ sum_j^k=\sum_{i=0}^{d}w_{ij}x_i^k$$

梯度：

$$\frac{\partial Cost}{\partial w_{ij}}=\sum_{k=1}^{n}\frac{\partial Cost}{\partial sum_j^k}\frac{\partial sum_j^k}{\partial w_{ij}}=\sum_{k=1}^{n}-(t_j^k-f(sum_j^k))f'(sum_j^k)x_i^k$$

权重修正量：

$$\triangle w_{ij}=-\lambda\frac{\partial Cost}{\partial w_{ij}}=\lambda\sum_{k=1}^{n}\delta_j^k x_i^k$$
$$\delta_j^k=f'(sum_j^k)(t_j^k-z_j^k)$$

其中，δ_j^k是误差(sensitivity)

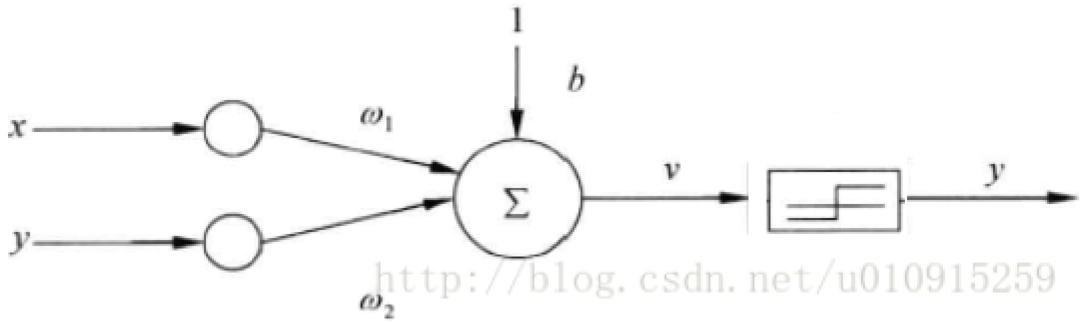
二、根据逻辑“与”功能的真值表，利用单层神经网络（以符号函数作为激励函数）来实现该功能。其中学习率、阈值（即偏置）、初始权重自己设定，“真”用“1”表示，“假”用“0”表示。

逻辑与的真值表：

k	x_1	x_2	T
1	0	0	-1(0)
2	0	1	-1(0)
3	1	0	-1(0)
4	1	1	1

逻辑“与”的真值表

网络结构图：



激励函数为符号函数时的梯度更新策略：

$$\triangle w_{ij} = \lambda(t_j^k - z_j^k)x_i^k = \lambda\delta_j^k x_i^k$$

$$w_{ij} = w_{ij} + \triangle w_{ij}$$

设权重初始值$w_1 = 0.5$,$w_2 = 0.5$, $b = 0.75$

学习率$\lambda = 0.02$

输入（0,0,1,-1）(解释：前两维为x_1 x_2,第三维的1为偏置项对应的输入，设为1，第四维是真实值)前向传播为：

$$0.5 * 0 + 0.5 * 0 + 1 * 0.75 = 0.75 > 0$$

预测值为：1

真实值为：-1

更新：$w_1 = 0.5$,$w_2 = 0.5$, $b = 0.75 + 0.02 * (-2) * 1 = 0.71$

输入(0,1,1,-1)，前向传播为：

$$0.5 * 0 + 0.5 * 1 + 0.71 * 1 = 1.21 > 0$$

预测值为：1

真实值为：-1

更新：$w_1 = 0.5$，$w_2 = 0.5 + 0.02 * (-2) * 1 = 0.46$，$b = 0.71 + 0.02 * (-2) * 1 = 0.67$

经过几轮的手动输入数据，w_1, w_2, b的数值趋于稳定：

W:[0.5 0.46 0.67]W:[0.46 0.46 0.63]W:[0.46 0.46 0.63]W:[0.46 0.46 0.59]W:[0.46 0.42 0.55] W:[0.42 0.42 0.51]W:[0.42 0.42 0.51]W:[0.42 0.42 0.47]W:[0.42 0.38 0.43]W:[0.38 0.38 0.39] W:[0.38 0.38 0.39]W:[0.38 0.38 0.35]W:[0.38 0.34 0.31]W:[0.34 0.34 0.27]W:[0.34 0.34 0.27] W:[0.34 0.34 0.23]W:[0.34 0.3 0.19]W:[0.3 0.3 0.15]W:[0.3 0.3 0.15]W:[0.3 0.3 0.11] W:[0.3 0.26 0.07]W:[0.26 0.26 0.03]W:[0.26 0.26 0.03]W:[0.26 0.26 -0.01]W:[0.26 0.22 -0.05] W:[0.22 0.22 -0.09]W:[0.22 0.22 -0.09]W:[0.22 0.22 -0.09]W:[0.22 0.18 -0.13]W:[0.18 0.18 -0.17] W:[0.18 0.18 -0.17]W:[0.18 0.18 -0.17]W:[0.18 0.14 -0.21]W:[0.18 0.14 -0.21]W:[0.18 0.14 -0.21] W:[0.18 0.14 -0.21]W:[0.18 0.14 -0.21]W:[0.18 0.14 -0.21]W:[0.18 0.14 -0.21]W:[0.18 0.14 -0.21] W:[0.18 0.14 -0.21]W:[0.18 0.14 -0.21]W:[0.18 0.14 -0.21]W:[0.18 0.14 -0.21]W:[0.18 0.14 -0.21]

三、与上一题条件类似，实现逻辑“异或”功能

逻辑异或的真值表：

k	x_1	x_2	T
1	0	0	-1(0)
2	0	1	1
3	1	0	1
4	1	1	-1(0)

逻辑“异或”的真值表

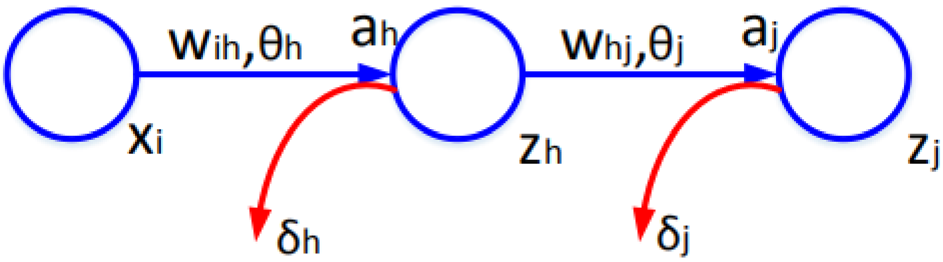
网络结构图与权重、学习率与上述一致

经过几轮的手动输入数据，w_1, w_2, b的数值如下所示：

W:[0.5 0.5 0.75]W:[0.5 0.5 0.55]W:[0.5 0.5 0.55]W:[0.5 0.5 0.55]W:[0.3 0.3 0.35] W:[0.3 0.3 0.15]W:[0.3 0.3 0.15]W:[0.3 0.3 0.15]W:[0.1 0.1 -0.05]W:[0.1 0.1 -0.05] W:[0.1 0.1 -0.05]W:[0.1 0.1 -0.05]W:[-0.1 -0.1 -0.25]W:[-0.1 -0.1 -0.25]W:[-0.1 0.1 -0.05] W:[0.1 0.1 0.15]W:[-0.1 -0.1 -0.05]W:[-0.1 -0.1 -0.05]W:[-0.1 0.1 0.15]W:[-0.1 0.1 0.15] W:[-0.3 -0.1 -0.05]W:[-0.3 -0.1 -0.05]W:[-0.3 0.1 0.15]W:[-0.1 0.1 0.35]W:[-0.3 -0.1 0.15] W:[-0.3 -0.1 -0.05]W:[-0.3 0.1 0.15]W:[-0.1 0.1 0.35]W:[-0.3 -0.1 0.15]W:[-0.3 -0.1 -0.05] W:[-0.3 0.1 0.15]W:[-0.1 0.1 0.35]W:[-0.3 -0.1 0.15]W:[-0.3 -0.1 -0.05]W:[-0.3 0.1 0.15] W:[-0.1 0.1 0.35]W:[-0.3 -0.1 0.15]W:[-0.3 -0.1 -0.05]W:[-0.3 0.1 0.15]W:[-0.1 0.1 0.35] W:[-0.3 -0.1 0.15]W:[-0.3 -0.1 -0.05]W:[-0.3 0.1 0.15]W:[-0.1 0.1 0.35]W:[-0.3 -0.1 0.15] W:[-0.3 -0.1 -0.05]W:[-0.3 0.1 0.15]W:[-0.1 0.1 0.35]W:[-0.3 -0.1 0.15]W:[-0.3 -0.1 -0.05] W:[-0.3 0.1 0.15]W:[-0.1 0.1 0.35]W:[-0.3 -0.1 0.15]W:[-0.3 -0.1 -0.05]W:[-0.3 0.1 0.15] W:[-0.1 0.1 0.35]W:[-0.3 -0.1 0.15] W:[-0.3 -0.1 -0.05]W:[-0.3 0.1 0.15]W:[-0.1 0.1 0.35]W:[-0.3 -0.1 0.15]W:[-0.3 -0.1 -0.05] W:[-0.3 0.1 0.15]W:[-0.1 0.1 0.35]W:[-0.3 -0.1 0.15]W:[-0.3 -0.1 -0.05]W:[-0.3 0.1 0.15] W:[-0.1 0.1 0.35]W:[-0.3 -0.1 0.15]W:[-0.3 -0.1 -0.05]W:[-0.3 0.1 0.15]W:[-0.1 0.1 0.35]

经过几轮的梯度迭代，w_1, w_2, b的数值并没有趋于稳定，说明单层神经网络并不能划分线性不可分的数据。

四、独立完成右侧网络模型的BP算法推导。



$$a_h = \sum_{i=1}^{d} w_{ih} x_i + \theta_h, z_h = f(a_h)$$
$$a_j = \sum_{h=1}^{n_H} w_{hj} z_h + \theta_j, z_j = f(a_j)$$

损失函数:

$$Cost = \frac{1}{2}\sum_j (t_j - z_j)^2$$

隐含层-输出层:

$$\begin{aligned}
&\frac{\partial Cost}{\partial w_{hj}} = \frac{\partial Cost}{\partial a_j}\frac{\partial a_j}{\partial w_{hj}} = \delta_j z_h \\
&\delta_j = \frac{\partial Cost}{\partial a_j} = \frac{\partial Cost}{\partial z_j}\frac{\partial z_j}{\partial a_j} = -(t_j - z_j) f'(a_j) \\
&\frac{\partial Cost}{\partial \theta_j} = \frac{\partial Cost}{\partial a_j}\frac{\partial a_j}{\partial \theta_j} = \delta_j
\end{aligned}$$

输入层-隐含层:

$$\begin{aligned}
&\frac{\partial Cost}{\partial w_{ih}} = \frac{\partial Cost}{\partial a_h}\frac{\partial a_h}{\partial w_{ih}} = \delta_h x_i \\
&\delta_h = \frac{\partial Cost}{\partial a_h} = \sum_{j=1}^{c} \frac{\partial Cost}{\partial a_j}\frac{\partial a_j}{\partial z_h}\frac{\partial z_h}{\partial a_h} = \sum_{j=1}^{c} \delta_j w_{hj} f'(a_h) \\
&\frac{\partial Cost}{\partial \theta_h} = \frac{\partial Cost}{\partial a_h}\frac{\partial a_h}{\partial \theta_h} = \delta_h
\end{aligned}$$