5.1 试述将线性函数 $f(x) = w^Tx$ 用作神经元激活函数的缺陷？

如果激活函数是一个线性函数，那么无论多少层网络，都表示为一层线性网络。如果用题目 f(x) 做激活函数，无论多少层神经网络都退化成了线性回归，达不到“激活”与“筛选”的目的。

5.2 试述使用图 5.2(b)激活函数的神经元与对率回归的联系

对率回归，是使用 Sigmoid 函数作为联系函数时的广义线性模型。单位阶跃函数不连续，难以求导，所以用对数几率函数代替。使用 Sigmoid 激活函数，每个神经元几乎和对率回归相同，只不过对率回归在 [sigmoid(x)>0.5] 时输出为 1，而神经元直接输出 [sigmoid(x)] 。

5.3 对于图 5.7 中的 V_{ih},试推导出 BP 算法中的更新公式(5.13)

5.3

已知网络在(x_k, y_k)上均方误差为：

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

给定学习率η，基于梯度下降有：

$$\Delta v_{ih} = -\eta \frac{\partial E_k}{\partial v_{ih}}.$$

$$\frac{\partial E_k}{\partial v_{ih}} = \sum_{j=1}^{l} \frac{\partial E_k}{\partial \hat{y}_j^k}\cdot\frac{\partial \hat{y}_j^k}{\partial \beta_j}\cdot\frac{\partial \beta_j}{\partial b_h}\cdot\frac{\partial b_h}{\partial \alpha_h}\cdot\frac{\partial \alpha_h}{\partial v_{ih}}.$$

$$g_j = -\frac{\partial E_k}{\partial \hat{y}_j^k}\cdot\frac{\partial \hat{y}_j^k}{\partial\beta_j} = \hat{y}_j^k(1-\hat{y}_j^k)(y_j^k-\hat{y}_j^k)$$

$$\frac{\partial\alpha_h}{\partial v_{ih}} = x_i \qquad \frac{\partial\beta_j}{\partial b_h} = w_{hj} \qquad \frac{\partial b_h}{\partial\alpha_h} = f'(\alpha_h-\gamma_h)$$

$$\frac{\partial E_k}{\partial v_{ih}} = \sum_{j=1}^{l} -g_j\cdot w_{hj}\cdot f'(\alpha_h-\gamma_h)\cdot x_i = -e_h x_i$$

其中$e_h = g_j w_{hj}$.

$$\therefore \Delta v_{ih} = \eta e_h x_i$$

5.4 试述式(5.6)中的学习率的取值对神经网络训练的影响

学习率 $\eta\in(0,1)$控制算法每一轮迭代中的更新步长，学习率 η太小，则每次的更新量会太小，这使得迭代次数增多；反之，如果学习率 η太大，则会出现震荡情况，即在最小值附近来回波动，导致算法无法收敛。