1.For the multiplelayer neural network model below,show that nonlinearity will not be achieved if the activation function f(.) is chosen as a linear function.

$$g_k(\mathbf{x}) \equiv y_k = f\left(\sum_j w_{kj} f\left(\sum_i w_{ji}x_i + w_{j0}\right) + w_{k0}\right),$$

Assume $f(x) = ax + b$,then we have

$$\Rightarrow f\left(\sum_i w_{ji}x_i + w_{j0}\right)$$
$$= \sum_i a w_{ji} x_i + (a w_{j0} + b).$$
$$\Rightarrow \sum_j w_{kj} f\left(\sum_i w_{ji}x_i + w_{j0}\right) + w_{k0}.$$
$$= \sum_i \sum_j a w_{kj} w_{ji} x_i + \cancel{\sum_j w_{kj}} \sum_j w_{kj}(a w_{j0} + b). + w_{k0}.$$
$$\Rightarrow f\left(\sum_j w_{kj} f\left(\sum_i w_{ji}x_i + w_{j0}\right) + w_{k0}\right).$$
$$= \sum_i \sum_j a^2 w_{kj} w_{ji} x_i + \sum_j a w_{kj}(a w_{j0} + b) + (a w_{k0} + b)$$
$$= y_k \equiv g_k(\vec{x}),\ \vec{x} \in R^n$$

Since function g(.) depicts a hyperplane in R^n,it is still a linear function.Therefore, nonlinearity will not be achieved if the activation function f(.) is chosen as a linear function.

2. Consider a single neuron with Sigmoid activation function $\mathbf{s(z) = 1/(1 + e^{-z})}$. The input of this neuron is $\mathbf{X = (x_0, \dots, x_n)^T}$ and the output is $\boldsymbol{y = s(W^T X)}$, whose weight vector being $\mathbf{W = (w_0, \dots, w_n)^T}$. The error function is $\mathbf{E = 0.5(g - y)^2}$, where g is the true label of samples.

(1)Write the weight-updating formula (Denote the learning rate as λ)

(2) Initially, the weight vector $\mathbf{W = (0.5, 1, 1)^T}$. If $\mathbf{X = (1, 2, 0.5)^T}$, g=1, λ=0.1. Write the new values of weight vector updated by one-step error back propagation.

(1) $\frac{\partial L}{\partial w} = X^T \otimes -(g-y)\cdot S(z)(1-S(z))$.

$= X^T \otimes (y-g)\cdot y(1-y)$

$\Rightarrow W_{new} = w - \lambda \cdot \frac{\partial L}{\partial w}$.

(2) Forward Propagation:

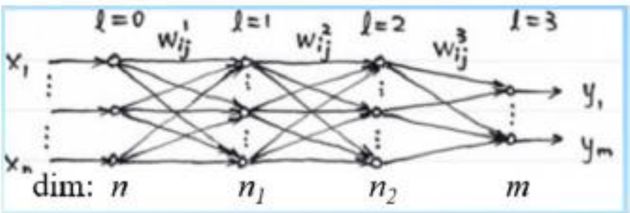
$y = \frac{1}{1+e^{-w^T x}} = 0.95$;

$\Rightarrow (y-g)\,y(1-y) \approx -2.375\times 10^{-3}$

After BP, the new weights are:

$W_{new} = (0.5 + 1.19\times 10^{-4},\ 1+4.75\times 10^{-4},\ 1+2.38\times 10^{-4})^T$.

3.(Optional)For a 4-layered MLP(with 3 hidden layers),derive the BP algorithm one by one layer and write down the pseudo-code for the training procedure.



The procedure of forward propagation has been illustrated in the left picture.

X_i, $\quad i = 1, 2, \cdots n$

$\downarrow$ Layer 0

$O_{0i} = f_0(X_i)$.

$\downarrow$ Layer 1

$net_{1h} = \sum_i w_{ih}^1 O_{0i}$, $\quad h = 1, 2, \cdots n_1$

$O_{1h} = f_1(net_{1h})$

$\downarrow$ Layer 2

$net_{2j} = \sum_h w_{hj}^2 O_{1h}$, $\quad j = 1, 2, \cdots n_2$

$O_{2j} = f_2(net_{2j})$

$\downarrow$ Layer 3

$net_{3k} = \sum_j w_{jk} O_{2j}$, $\quad k = 1, 2, \cdots m$

$\hat{y}_k \equiv O_{3k} = f_3(net_{3k})$.

$E = \sum_k \frac{1}{2}(y_k - \hat{y}_k)^2$

Pay attention to these math notations please,since the derivation of backward propagation algorithm will be based on the usage of them in the right picture.

$$\Rightarrow \frac{\partial E}{\partial w_{jk}^3} = \frac{\partial E}{\partial net_{3k}} \cdot \frac{\partial net_{3k}}{\partial w_{jk}^3}$$

$$= \delta_3 \cdot O_{2j}$$

$$\delta_3 = \frac{\partial E}{\partial net_{3k}} = \frac{\partial E}{\partial \hat{y}_k} \cdot \frac{\partial \hat{y}_k}{\partial net_{3k}}$$

$$= -(y_k - \hat{y}_k) \cdot f_3'(net_{3k})$$

$$\therefore \frac{\partial E}{\partial w_{jk}^3} = -(y_k - \hat{y}_k) f_3'(net_{3k})$$

$$\Rightarrow \frac{\partial E}{\partial w_{hj}^2} = \frac{\partial E}{\partial net_{2j}} \cdot \frac{\partial net_{2j}}{\partial w_{hj}^2}.$$

$$= \delta_2 O_{1h}.$$

$$\delta_2 = \frac{\partial E}{\partial net_{2j}} = \sum_k \frac{\partial E}{\partial net_{3k}} \cdot \frac{\partial net_{3k}}{\partial O_{2j}} \cdot \frac{\partial O_{2j}}{\partial net_{2j}}$$

$$= \sum_k \delta_3 \cdot w_{jk}^3 \cdot f'(net_{2j}) = \sum_k -(y_k - \hat{y}_k) f_3'(net_{3k}) \overset{O_{2j}}{\vee} w_{jk}^3 \cdot f'(net_{2j}).$$

$$\therefore \cancel{\frac{\partial E}{\partial w_{jk}^3}} \frac{\partial E}{\partial w_{hj}^2} = \sum_k -(y_k - \hat{y}_k) \cdot f_3'(net_{3k})\, O_{2j} \cdot w_{jk}^3 \cdot f'(net_{2j})\, O_{1h}.$$

$$\frac{\partial E}{\partial w_{hj}^2} = \sum_k -(y_k - \hat{y}_k) f_3'(net_{3k}) f'(net_{2j})\, w_{jk}^3\, O_{2j}\, O_{1h}.$$

$$\Rightarrow \frac{\partial E}{\partial w_{ih}^1} = \frac{\partial E}{\partial net_{1h}} \frac{\partial net_{1h}}{\partial w_{ih}^1}$$

$$= \delta_1 \cdot O_{0i}$$

$$\delta_1 = \frac{\partial E}{\partial net_{1h}} = \sum_j \frac{\partial E}{\partial net_{2j}} \cdot \frac{\partial net_{2j}}{\partial O_{1h}} \cdot \frac{\partial O_{1h}}{\partial net_{1h}}$$

$$= \sum_j \sum_k \delta_3 w_{jk}^3 f'(net_{2j}) \cdot w_{hj}^2 f_1'(net_{1h}).$$

$$= \sum_j \sum_k -(y_k - \hat{y}_k) \cdot f_3'(net_{3k})\, O_{2j}\, w_{jk}^3 f_2'(net_{2j}) \cdot w_{hj}^2 f_1'(net_{1h}).$$

$$\therefore \frac{\partial E}{\partial w_{ih}^1} = \sum_j \sum_k -(y_k - \hat{y}_k) f_3'(net_{3k}) \cdot f_2'(net_{2j}) \cdot f_1'(net_{1h})\, w_{jk}^3 w_{hj}^2\, O_{2j}\, O_{1h}\, O_{0i}$$