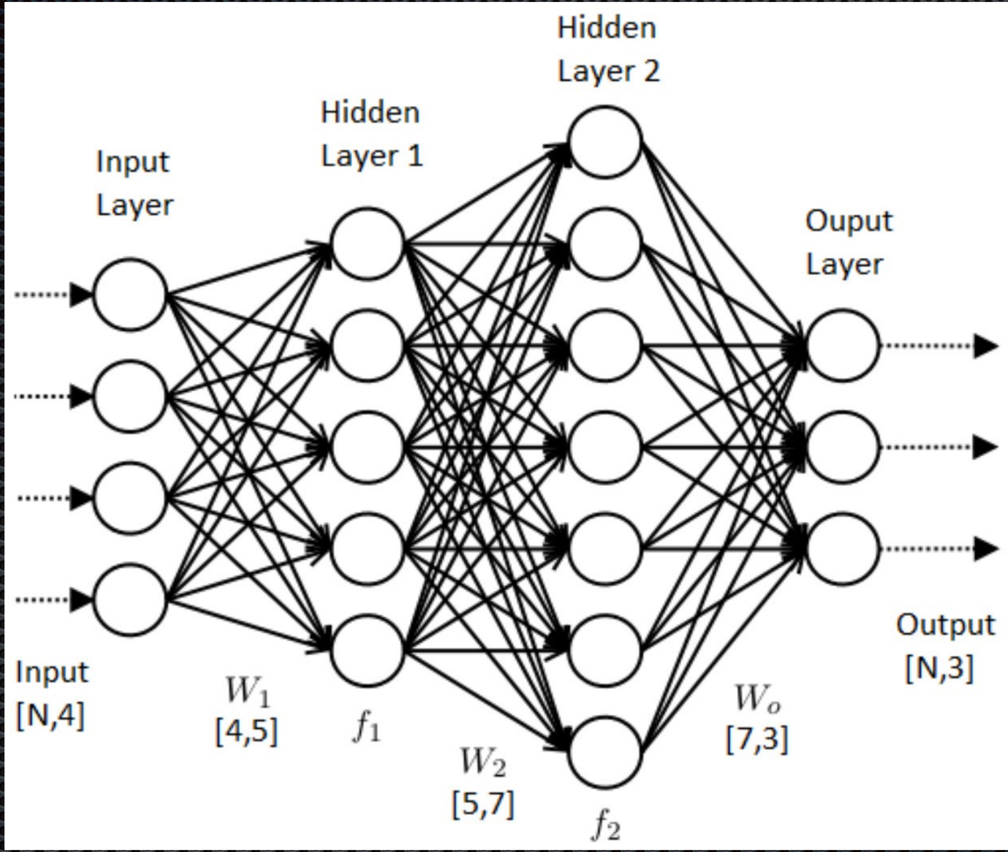
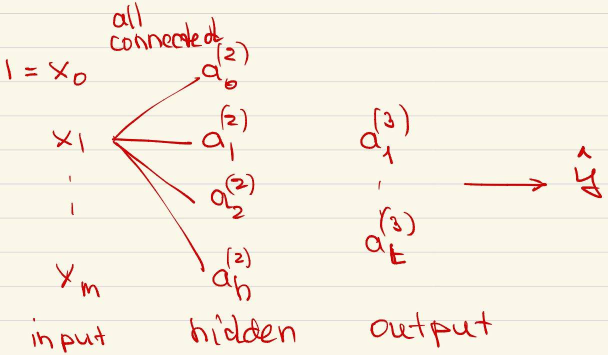
Artificial Neural Networks



E. Multi-layer perceptron (or ANN)

This 'machine' consists of an input layer of m units, a number n_h of hidden layers of h_k units, $k = 1, \ldots, n_h$ and an output layer of t units.

For simplicity, we will consider $n_h = 1$ and $h_1 = h$.



Weights : $w_{j,k}^{(\ell)}$

connecting units

k in layer ℓ with

j in layer $\ell+1$

Feed-forward network :

$$z_1^{(2)} = a_0^{(1)} w_{1,0}^{(1)} + \dots + a_m^{(1)} w_{1,m}^{(1)}$$

with
$$\begin{pmatrix} a_0^{(1)} \\ a_1^{(1)} \\ \vdots \\ a_m^{(1)} \end{pmatrix} = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_m \end{pmatrix}$$

$$a_1^{(2)} = \phi\left(z_1^{(2)}\right) \quad \text{with}$$

$$\phi(z) = \frac{1}{1+e^{-z}}$$

More general:

$$z_j^{(2)} = \sum_{k=0}^{m} a_k^{(1)} w_{j,k}^{(1)} \qquad j = 1, \ldots, h$$

or

$$\underline{z}^{(2)} = W^{(1)} \underline{a}^{(1)} \qquad W^{(1)}: h \times (m+1)$$

$$\underline{a}^{(2)} = \phi\left(\underline{z}^{(2)}\right) \qquad \underline{a}^{(1)}: (m+1) \times 1$$

$$\underline{z}^{(2)}, \underline{a}^{(2)}: h \times 1$$

Suppose, we have now N samples of training data, $i = 1, \ldots, N$, then

$$Z^{(2)} = W^{(1)} \left(A^{(1)}\right)^T$$

where $A^{(1)}: N \times m+1$

$Z^{(2)}: h \times N$

and $A^{(2)} = \phi\left(Z^{(2)}\right) \qquad A^{(2)}: h \times N$

componentwise

For the output layer (layer(3))
we can also write :

$$Z^{(3)} = W^{(2)} A^{(2)}$$

$$W^{(2)} : t \times h$$

$$A^{(2)} : h \times N$$

$$A^{(3)}, Z^{3} : t \times N$$

with $A^{(3)} = \phi\left(Z^{(3)}\right)$

<u>Output</u> : $\left(\hat{y}^{i}\right)_j = A^{(3)}_{ji}$

$j = 1, \dots, t$

$i = 1, \dots, N$

F. Training an ANN

Cost function (logistic)

To be compatible with notation of Raschka, let $a_j^i = A_{j,i}^{(3)}$, then the cost function is given by

$$J(\underline{w}) = -\sum_{i=1}^{N}\sum_{j=1}^{t}\Big\{ y_j^i \ln \phi(z_j^i) + $$

$$+ (1 - y_j^i) \ln \big[1 - \phi(z_j^i)\big]\Big\}$$

$$+ \underbrace{\frac{\lambda}{2} \sum_{l=1}^{L-1} \sum_{i=1}^{u_l} \sum_{j=1}^{u_{l+1}} w_{j,i}^{(l)}}_{\text{weight over all layers.}}$$

where also $z_j^i = Z_{j^i}^{(2)}$

Again the weights will be updated by the gradient of J, and hence we need.

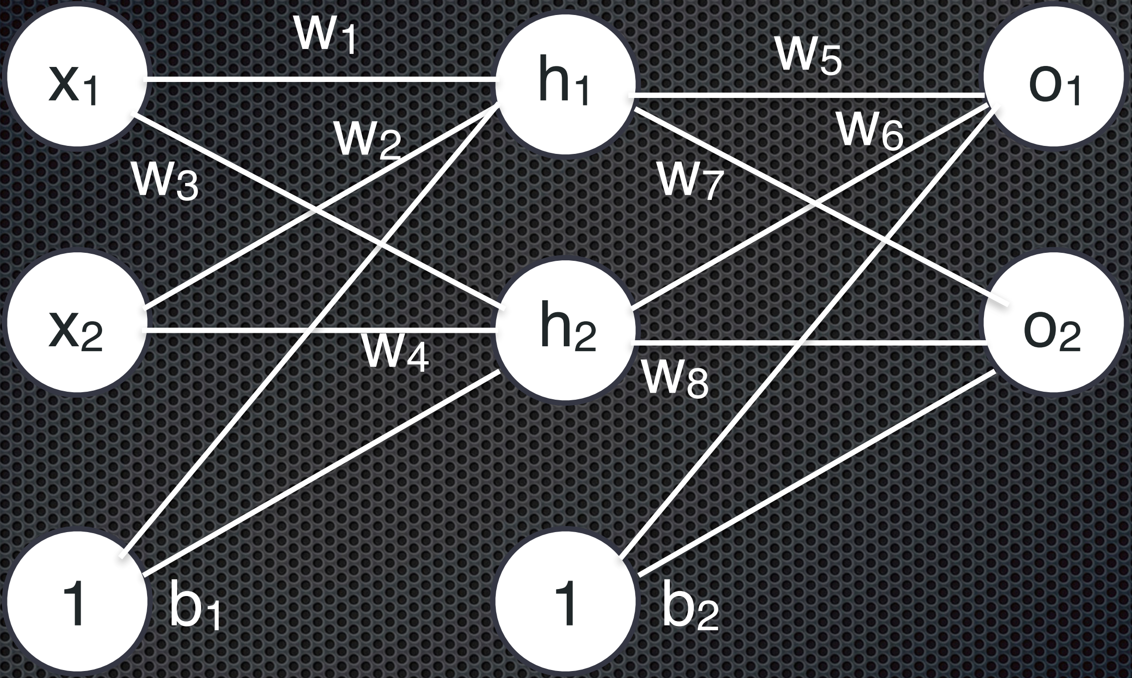
$$\frac{\partial}{\partial w_{j,k}^{(l)}} J\left(W^{(1)}, W^{(2)}\right)$$

where $W^{(1)}$ connects the hidden layer to the input layer and $W^{(2)}$ the hidden layer to the output layer. The weights will be updated through

$$W_{k+1}^{(l)} = W_k^{(l)} - \eta\, \Delta_{k+1}^{(l)},$$

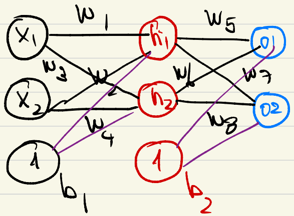
where $\Delta^{(l)}$ is determined from the so-called backpropagation algorithm

Example network with simpler cost function



$$J(\mathbf{w}) = \frac{1}{2}((y_1 - o_1)^2 + (y_2 - o_2)^2)$$

Example of backpropagation



data

y_1

y_2

Hidden : $z_1 = w_1 x_1 + w_2 x_2 + b_1$

$$h_1 = \phi(z_1) = \frac{1}{1+e^{-h_1}}$$

$$z_2 = w_3 x_1 + w_4 x_2 + b_1$$

$$h_2 = \phi(z_2) = \frac{1}{1+e^{-h_2}}$$

Output : $z_3 = w_5 h_1 + w_6 h_2 + b_2$

$$o_1 = \phi(z_3)$$

$$z_4 = w_7 h_1 + w_8 h_2 + b_2$$

$$o_2 = \phi(z_4)$$

$$\mathcal{J}(\underline{w}) = \frac{1}{2}\left\{ \left(y_1 - o_1 \right)^2 + \left(y_2 - o_2 \right)^2 \right\}$$

$$\underline{w} = \left(w_1, \dots, w_8, b_1, b_2 \right)^T$$

1) Consider first the weights of links connecting the hidden and the output layer, e.g. w_5.

$$\frac{\partial \mathcal{J}}{\partial w_5} = \frac{\partial \mathcal{J}}{\partial o_1} \frac{\partial o_1}{\partial z_3} \frac{\partial z_3}{\partial w_5}$$

$$\frac{\partial \mathcal{J}}{\partial o_1} = -\left(y_1 - o_1 \right)$$

$$\frac{\partial o_1}{\partial z_3} = \phi'(z_3) = \phi(z_3)\left(1 - \phi(z_3)\right)$$

$$= o_1 \left(1 - o_1 \right)$$

$$\frac{\partial z_3}{\partial w_5} = h_1$$

hence

$$\frac{\partial J}{\partial w_5} = -(y_1 - o_1)\, o_1 (1 - o_1)\, h_1$$

$$= \delta_{o_1} h_1$$

Hence, $\hat{w}_5 = w_5 - \eta \dfrac{\partial J}{\partial w_5}$ update

The same can be done for w_6, w_7, w_8.

2) Now consider the update for w_1:

$$\frac{\partial J}{\partial w_1} = \frac{\partial J}{\partial h_1} \frac{\partial h_1}{\partial z_1} \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial J}{\partial h_1} = \frac{\partial J}{\partial o_1} \cdot \frac{\partial o_1}{\partial h_1} + \frac{\partial J}{\partial o_2} \frac{\partial o_2}{\partial h_1}$$

$$\frac{\partial h_1}{\partial z_1} = h_1 (1 - h_1)$$

$$\frac{\partial z_1}{\partial w_1} = x_1$$

and $$\frac{\partial o_1}{\partial h_1} = o_1(1-o_1)\, w_5$$

$$\frac{\partial o_2}{\partial h_1} = o_2(1-o_2)\, w_7$$

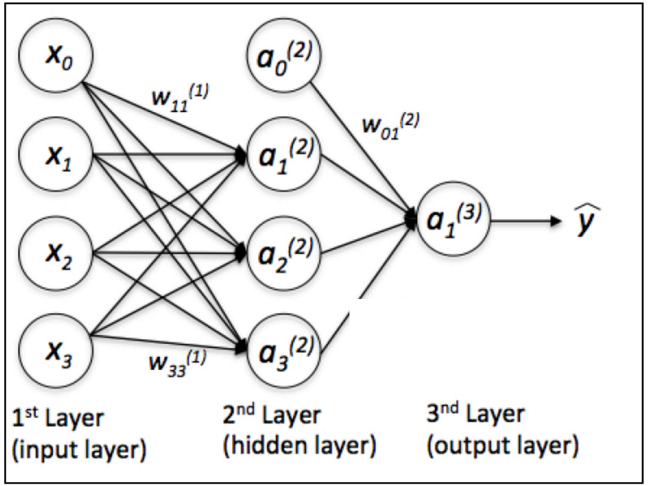
hence we can write:

$$\frac{\partial E}{\partial w_1} = \delta_{h_1} \cdot x_1 \quad , \text{ where}$$

$$\delta_{h_1} = -\left\{ (y_1 - o_1)\, o_1(1-o_1)\, w_5 + (y_2 - o_2)\, o_2(1-o_2)\, w_7 \right\} * h_1(1-h_1)$$

and this can also ~~be~~ done for the other weights w_2, w_3 and w_4.

Summary



$$a^{(1)} = \begin{bmatrix} a_0^{(1)} \\ a_1^{(1)} \\ \vdots \\ a_m^{(1)} \end{bmatrix} = \begin{bmatrix} 1 \\ x_1^{(i)} \\ \vdots \\ x_m^{(i)} \end{bmatrix}$$

$$z_1^{(2)} = a_0^{(1)} w_{1,0}^{(1)} + a_1^{(1)} w_{1,1}^{(1)} + \cdots + a_m^{(1)} w_{1,m}^{(1)}$$

$$a_1^{(2)} = \phi\left(z_1^{(2)}\right)$$

$w_{j,k}^{(l)}$ weight connecting unit j, level l+1 to unit k, level l

$$\phi(z) = \frac{1}{1+e^{-z}}$$

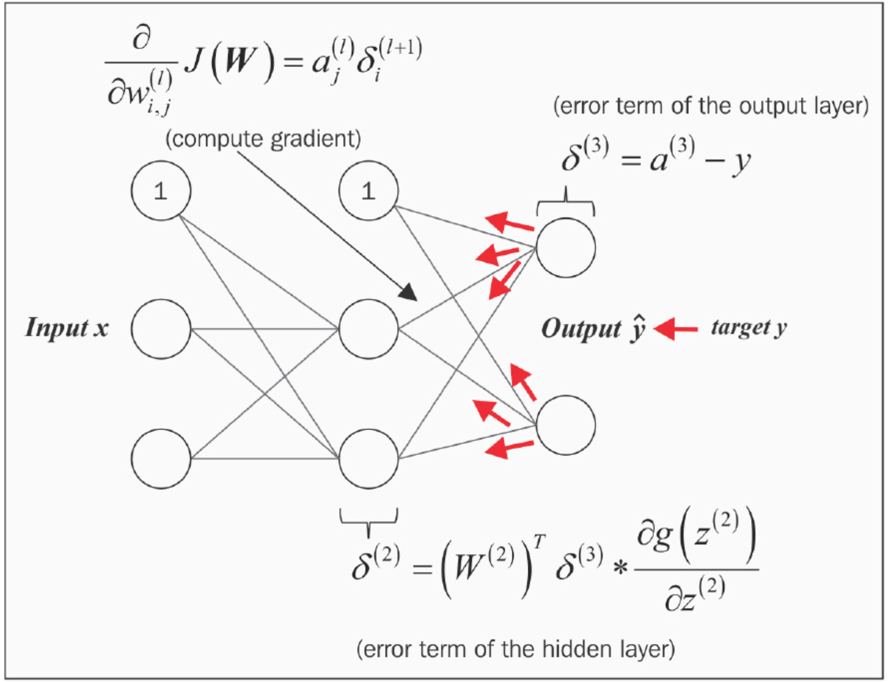
Cost function

$$J(\boldsymbol{w}) = -\left[\sum_{i=1}^{n}\sum_{j=1}^{m} y_j^{(i)} \log\left(\phi\left(z_j^{(i)}\right)\right) + \left(1 - y_j^{(i)}\right)\log\left(1 - \phi\left(z_j^{(i)}\right)\right)\right]$$

$$+\frac{\lambda}{2}\sum_{l=1}^{L-1}\sum_{i=1}^{u_l}\sum_{j=1}^{u_{l+1}}\left(w_{j,i}^{(l)}\right)^2$$

Task: $min_{\mathbf{w}} \ J(\mathbf{w})$

Solution: back propagation



$$\Delta^{(l)} = \Delta^{(l)} + \delta^{(l+1)} \left(A^{(l)} \right)^T$$

$$\mathbf{W}_{k+1}^{(l)} = \mathbf{W}_{k}^{(l)} - \eta \Delta^{(l)}$$

Exercise B4