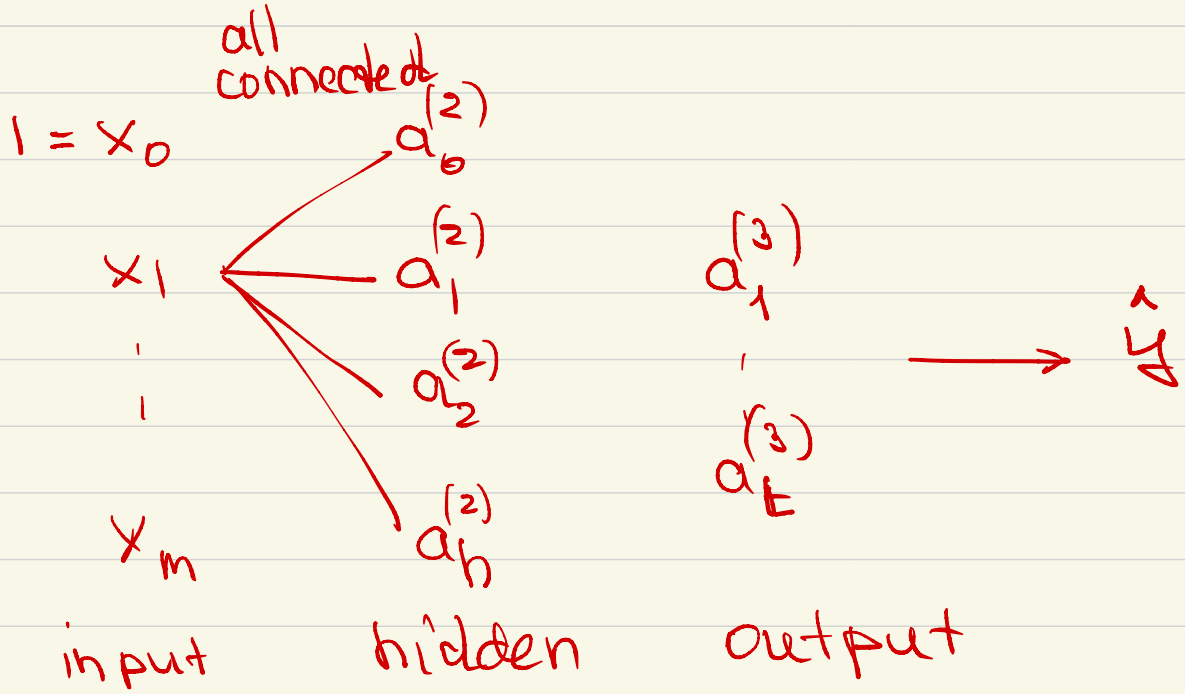
E. Multi-layer perceptron (or ANN)

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

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

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 $A^{(2)} = \phi(Z^{(2)})$ $\quad A^{(2)}: h \times N$

↑ 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 $A^{(3)} = \varphi(Z^{(3)})$

Output : $\left(\begin{matrix} 1 & i \\ y & \\ j & \end{matrix}\right)_{j,i} = A^{(3)}_{j,i}$

$j = 1, \ldots, t$

$i = 1, \ldots, N$

F. Training an ANN

Cost function (logistic)

To be compatible with notation of Raschka, let $a_j^i = A_{j\,1}^{(3)}$, then the cost function is given by

$$J(\underline{w}) = -\sum_{i=1}^{N}\sum_{j=1}^{t}\left\{ y_j^i \ln \phi\left(z_j^i\right) + \left(1 - y_j^i\right) \ln\left[1 - \phi\left(z_j^i\right)\right]\right\} + \frac{\lambda}{2}\sum_{l=1}^{L-1}\sum_{i=1}^{u_l}\sum_{j=1}^{u_{l+1}} w_{j,i}^{(l)}$$

weight over all layers.

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

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

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

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

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

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