DAT300 – Applied Deep Learning

Math from ANN

A multi-layer neural network architecture

- Data arrays

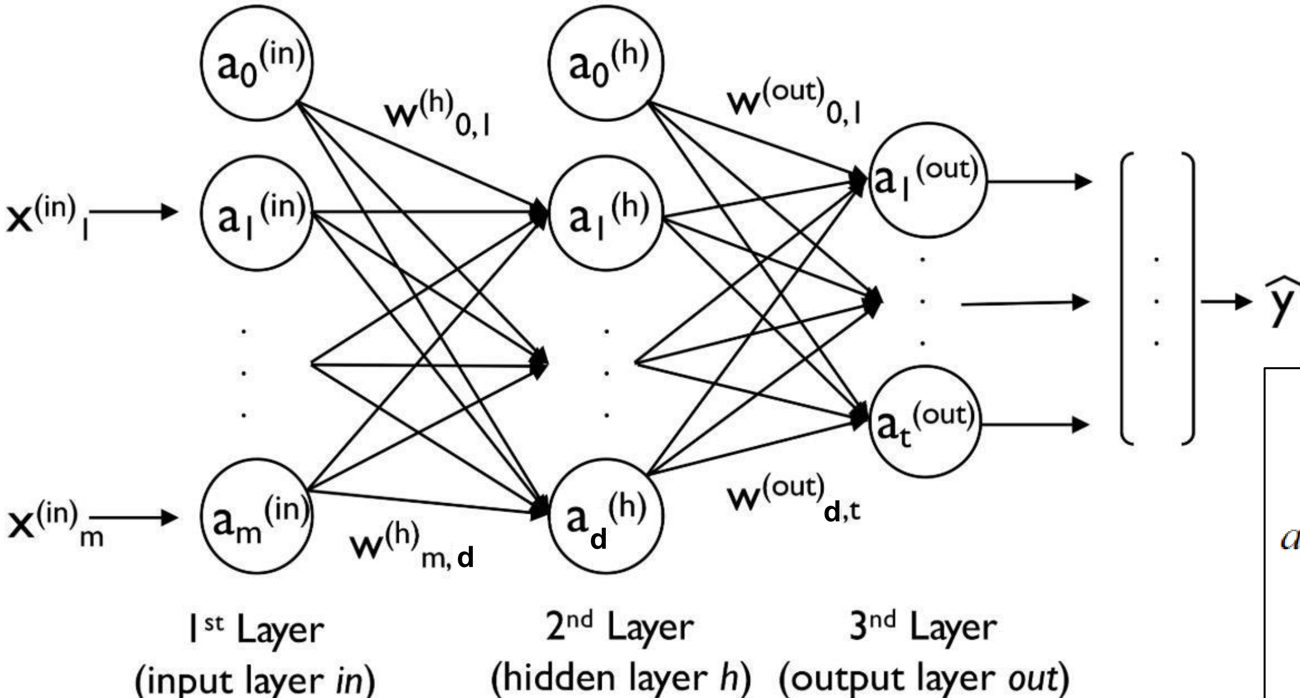
(n x m) (n x 1)

- Indexing

$i = 1, \ldots, n$ sample / instance index

$j = 0, \ldots, m$ feature / instance index

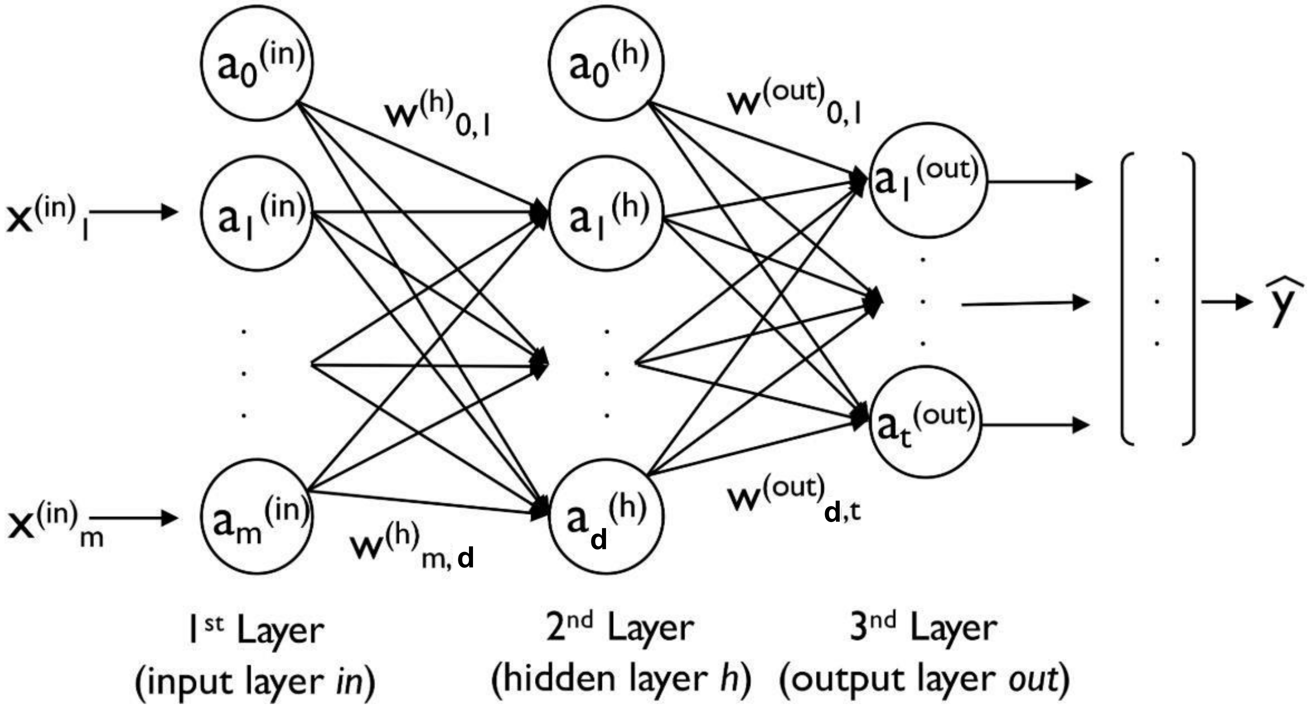
A multi-layer neural network architecture



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

(m x 1) + bias (m x 1) + bias

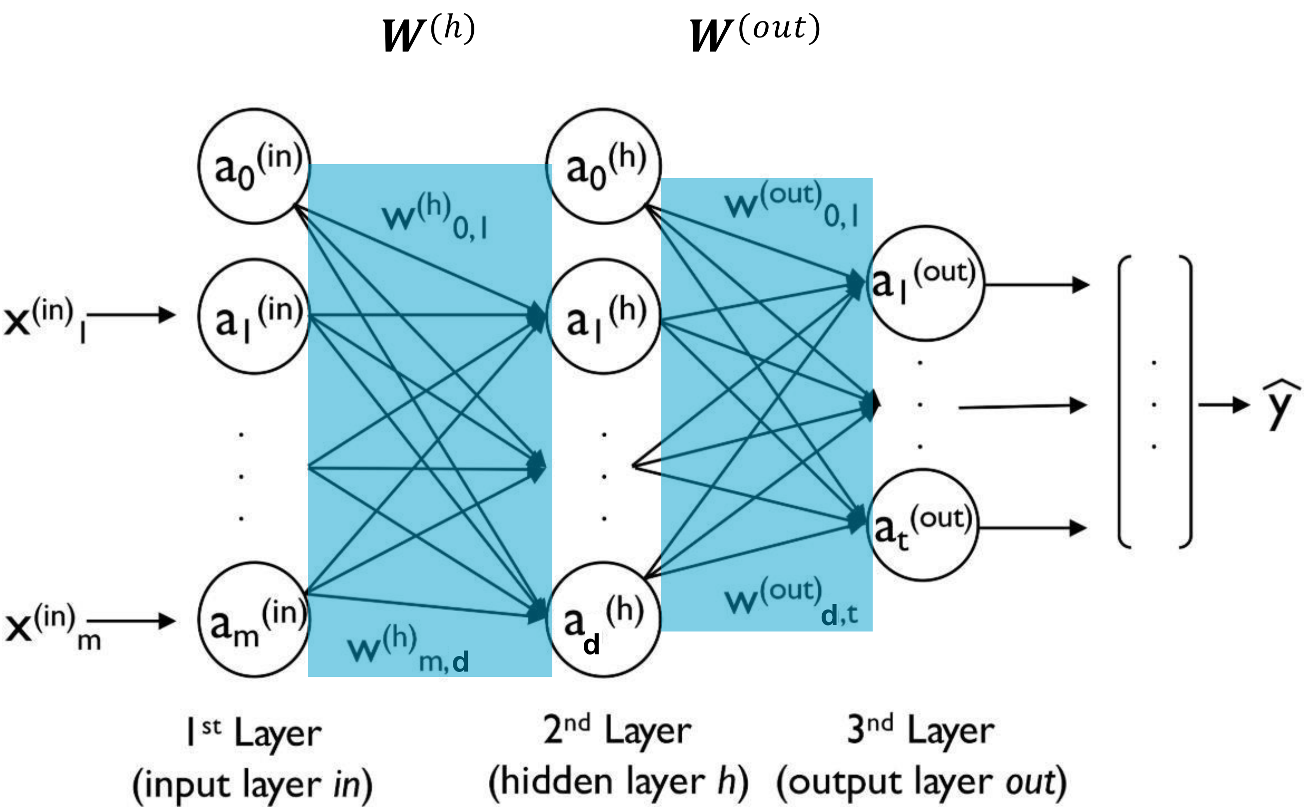
A multi-layer neural network architecture

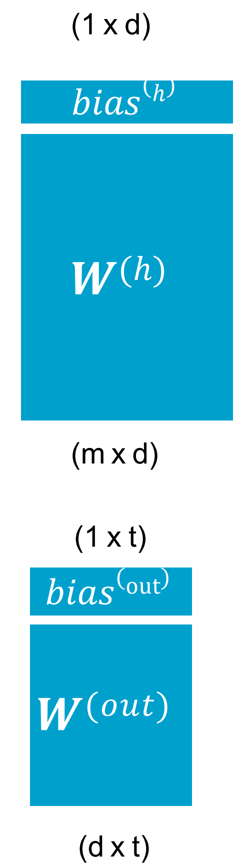


Example of model for three classes with t=3

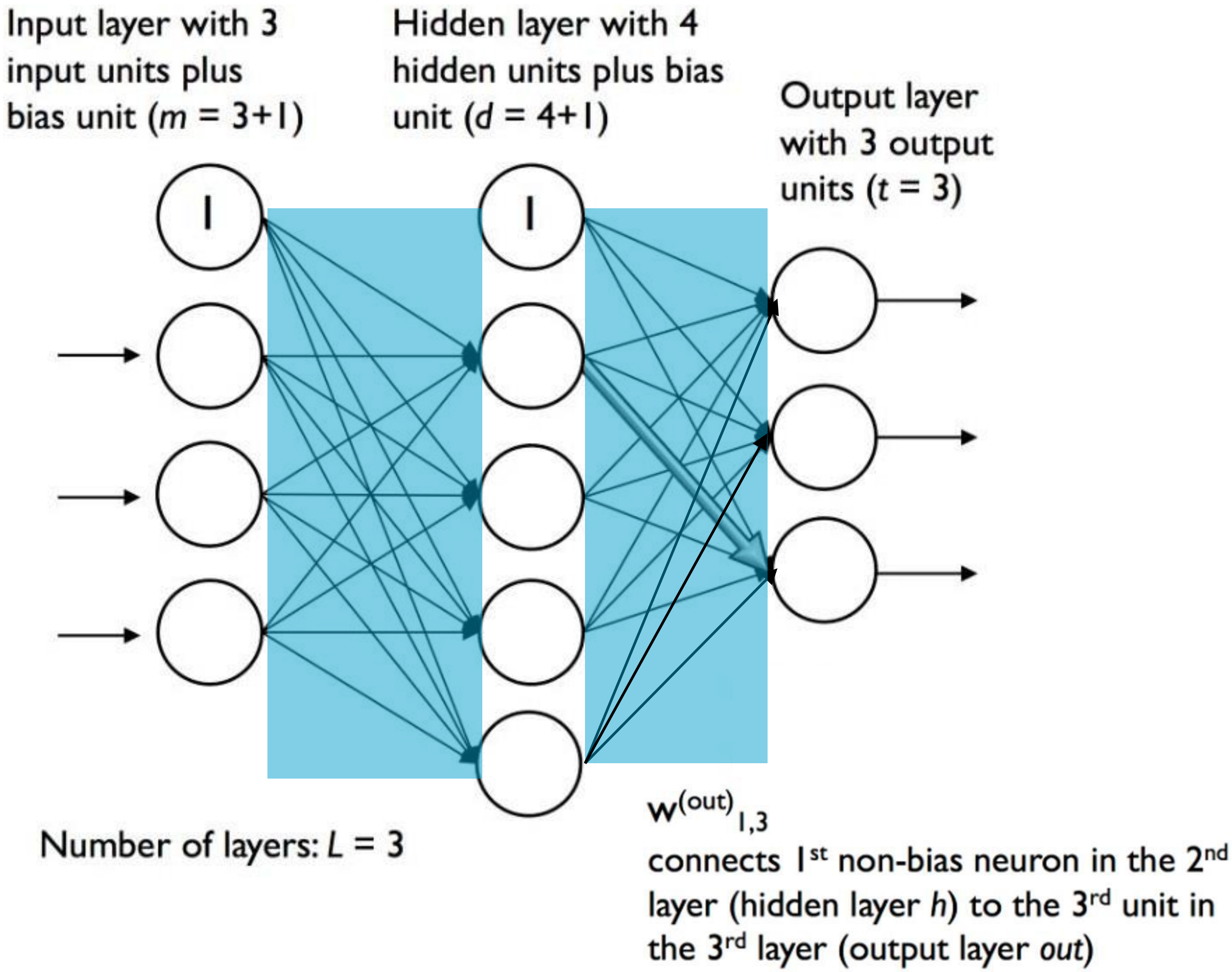
$$0 = \begin{bmatrix}1\\0\\0\end{bmatrix}, 1 = \begin{bmatrix}0\\1\\0\end{bmatrix}, 2 = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

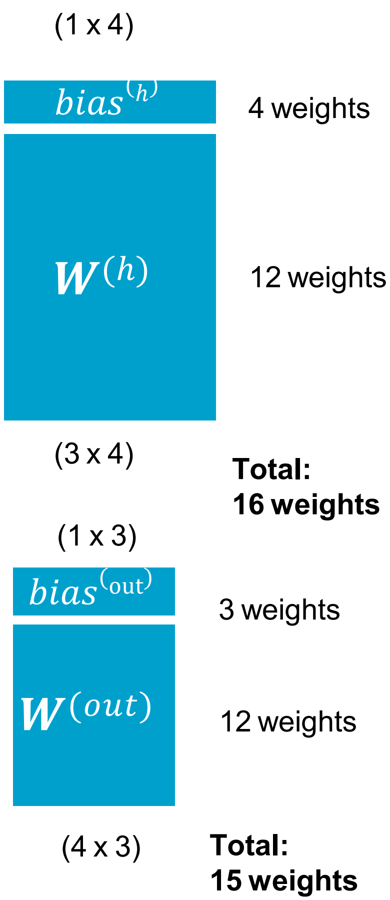
A multi-layer neural network architecture





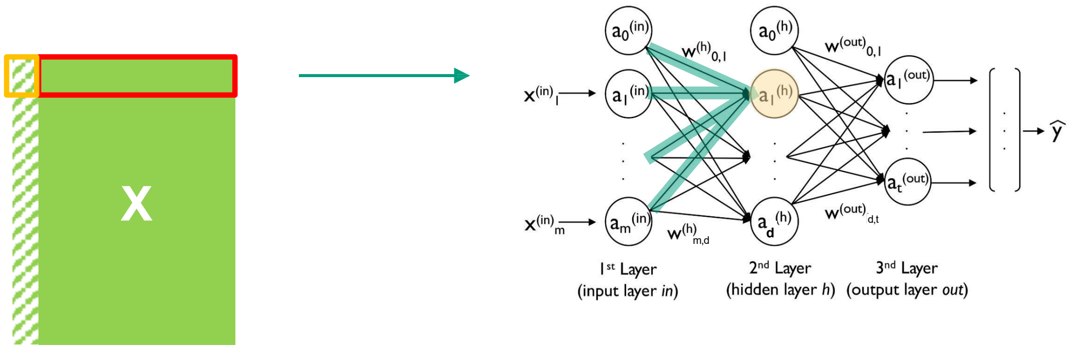
A 3-4-3 multi-layer perceptron





Need to train 16 + 15 = 31 weights in NN model

A 3-4-3 multi-layer perceptron



(n x m)

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

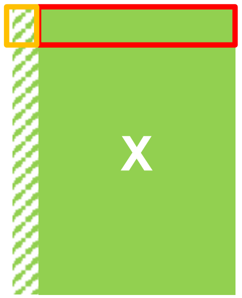
(1 x 1) (1 x 1) (1 x 1) (1 x 1) (1 x 1) (1 x 1) (1 x 1)

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

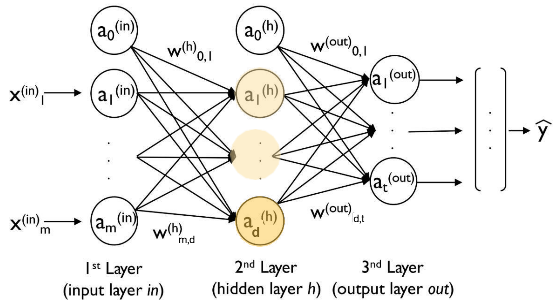
(1 x 1) (1 x 1)

Computations for one sample (row) x_i in $\boldsymbol{X}$ for one neuron in hidden layer

A 3-4-3 multi-layer perceptron



(n x m)



m: features
d: neurons
Samples: 1

$$\boldsymbol{z}^{(h)} = \boldsymbol{a}^{(in)}\boldsymbol{W}^{(h)}$$

(1 x d) (1 x m) (m x d)

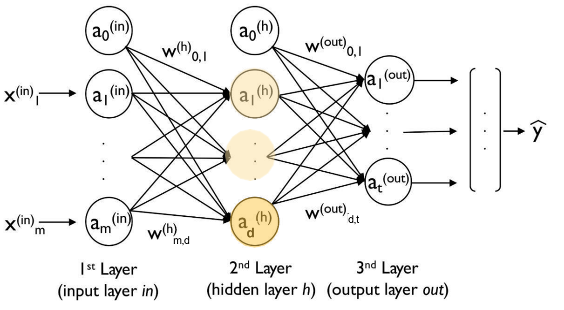
$$\boldsymbol{a}^{(h)} = \phi\left(\boldsymbol{z}^{(h)}\right)$$

(1 x d) (1 x d)

Computations for one sample (row) x_i in $\boldsymbol{X}$ for all neurons in hidden layer

A 3-4-3 multi-layer perceptron

(n x m)



m: features
d: neurons
Samples: n

$$\boldsymbol{Z}^{(h)} = \boldsymbol{A}^{(in)} \boldsymbol{W}^{(h)}$$

(n x d) (n x m) (m x d)

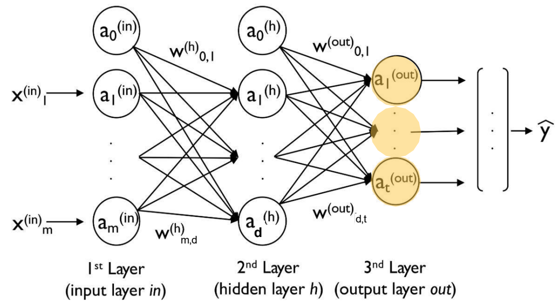
$$\boldsymbol{A}^{(h)} = \phi\left(\boldsymbol{Z}^{(h)}\right)$$

(n x d) (n x d)

Computations for all n samples (rows) x_i in $\boldsymbol{X}$ for all neurons in hidden layer

A 3-4-3 multi-layer perceptron

(n x m)



t: labels
d: neurons
Samples: n

Computations
for all n samples
(rows) x_i in $\boldsymbol{X}$
for all neurons in
output layer

$$\boldsymbol{Z}^{(out)} = \boldsymbol{A}^{(h)} \boldsymbol{W}^{(out)}$$

(n x t) (n x d) (d x t)

$$\boldsymbol{A}^{(out)} = \phi\left(\boldsymbol{Z}^{(out)}\right)$$

(n x t) (n x t)