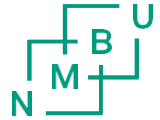


# DAT300 – Applied Deep Learning

Math from ANN

# A multi-layer neural network architecture



- Data arrays



$(n \times m)$



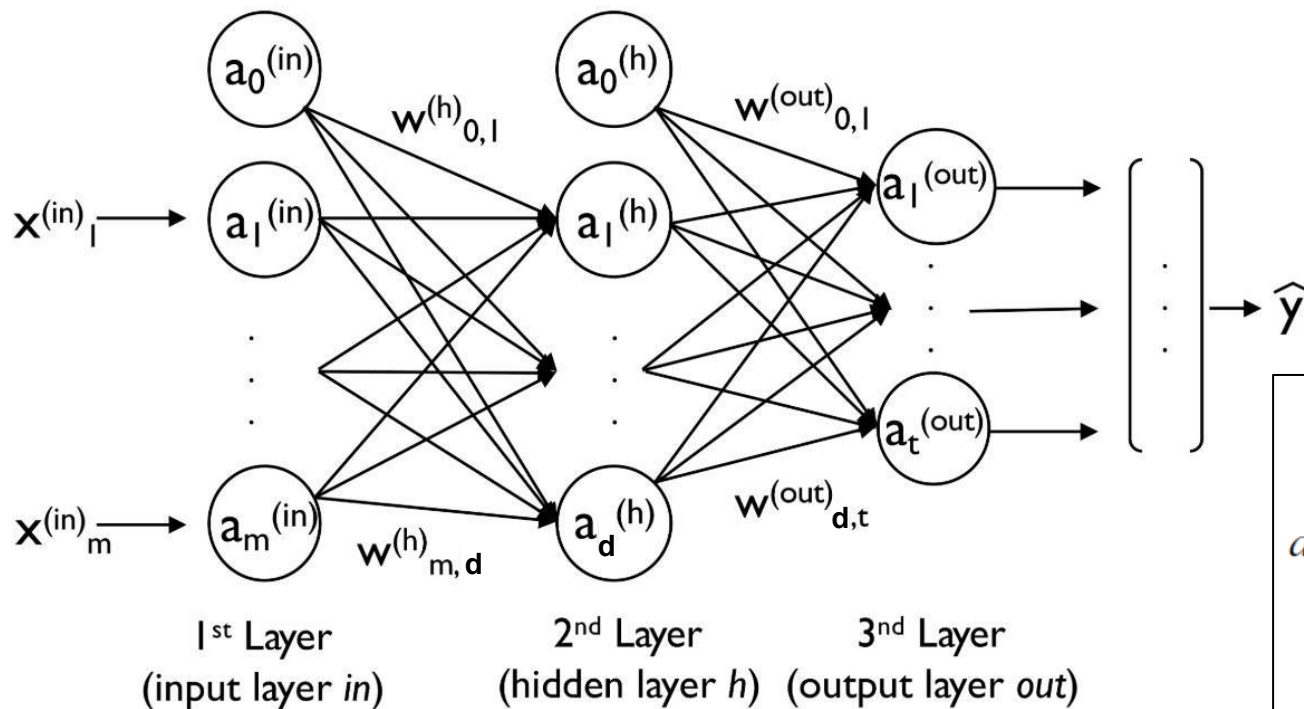
$(n \times 1)$

- Indexing

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

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

# A multi-layer neural network architecture

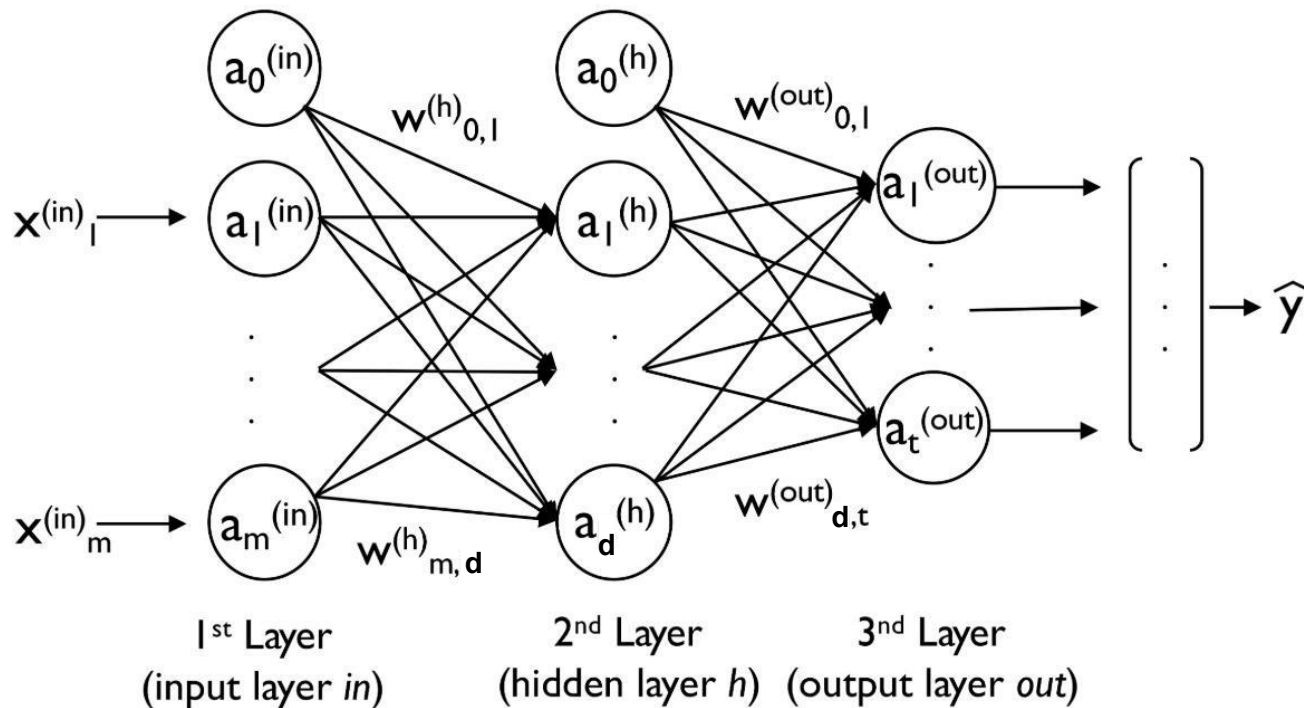
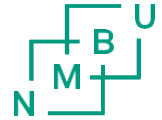


$$a^{(in)} = \begin{bmatrix} a_0^{(in)} \\ a_1^{(in)} \\ \vdots \\ a_m^{(in)} \end{bmatrix} = \begin{bmatrix} 1 \\ x_1^{(in)} \\ \vdots \\ x_m^{(in)} \end{bmatrix}$$

(m x 1)  
+ bias

(m x 1)  
+ bias

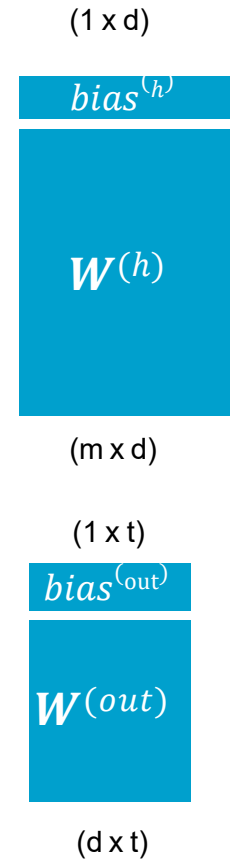
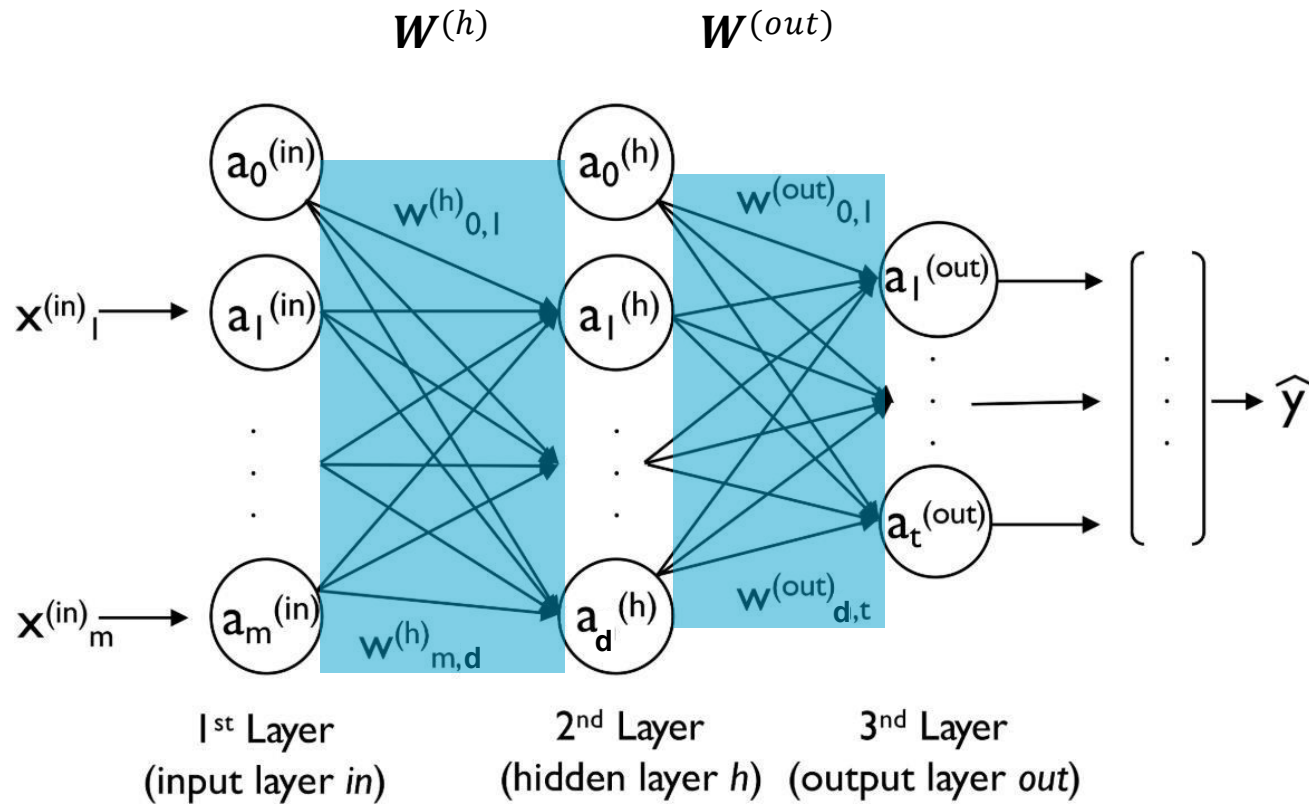
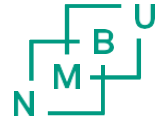
# A multi-layer neural network architecture



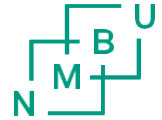
Example of model for three classes with  $t=3$

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

# A multi-layer neural network architecture



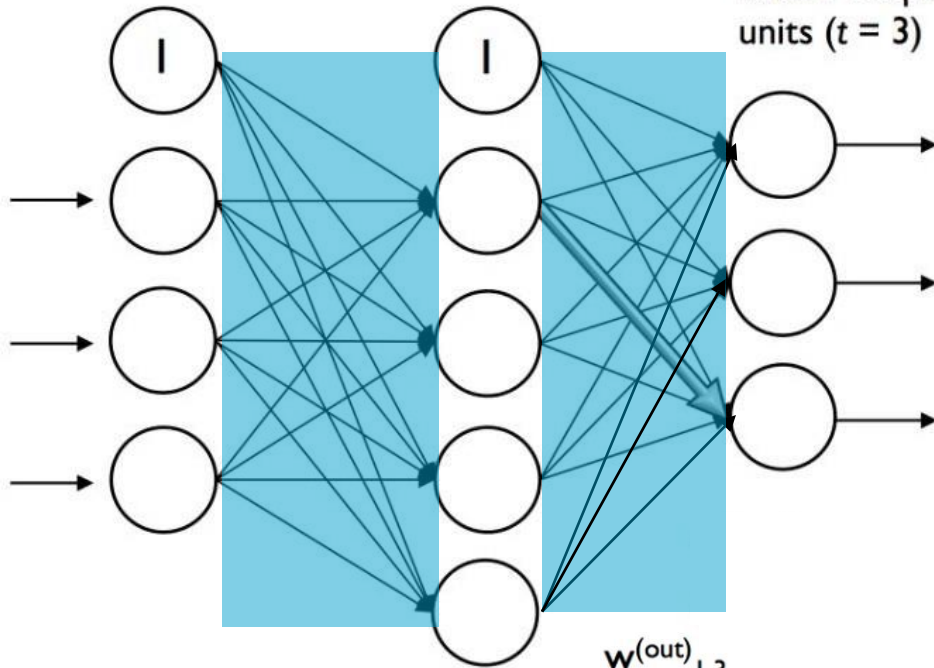
# A 3-4-3 multi-layer perceptron



Input layer with 3 input units plus bias unit ( $m = 3 + 1$ )

Hidden layer with 4 hidden units plus bias unit ( $d = 4 + 1$ )

Output layer with 3 output units ( $t = 3$ )

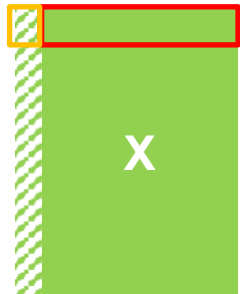
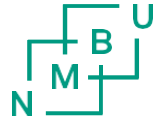


Number of layers:  $L = 3$

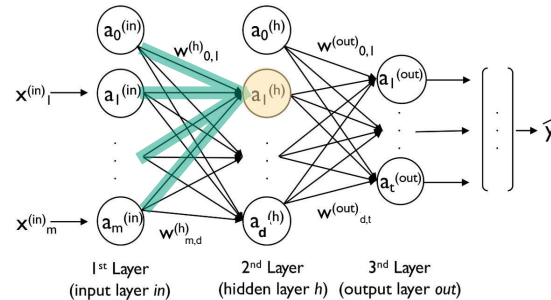
$w^{(out)}_{1,3}$   
connects 1<sup>st</sup> non-bias neuron in the 2<sup>nd</sup> layer (hidden layer  $h$ ) to the 3<sup>rd</sup> unit in the 3<sup>rd</sup> layer (output layer  $out$ )

$(1 \times 4)$	$bias^{(h)}$	4 weights
	$W^{(h)}$	12 weights
$(3 \times 4)$		<b>Total: 16 weights</b>
$(1 \times 3)$	$bias^{(out)}$	3 weights
	$W^{(out)}$	12 weights
$(4 \times 3)$		<b>Total: 15 weights</b>

# A 3-4-3 multi-layer perceptron



(n x m)

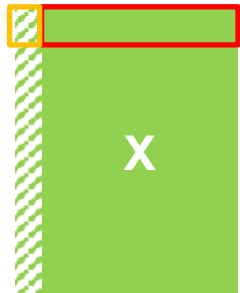
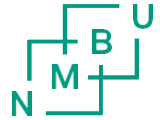


$$\begin{matrix}
 z_1^{(h)} & = & a_0^{(in)} w_{0,1}^{(h)} + a_1^{(in)} w_{1,1}^{(h)} + \dots + a_m^{(in)} w_{m,1}^{(h)} \\
 (1 \times 1) & & (1 \times 1) \quad (1 \times 1) & (1 \times 1) \quad (1 \times 1) & (1 \times 1) \quad (1 \times 1)
 \end{matrix}$$

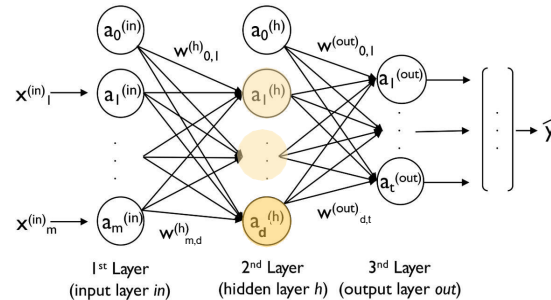
$$\begin{matrix}
 a_1^{(h)} & = & \phi \left( z_1^{(h)} \right) \\
 (1 \times 1) & & (1 \times 1)
 \end{matrix}$$

Computations for **one sample** (row)  $x_i$  in  $X$  for **one neuron** in hidden layer

# A 3-4-3 multi-layer perceptron



(n x m)



m: features  
d: neurons  
# Samples: 1

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

(1 x d)      (1 x m)      (m x d)

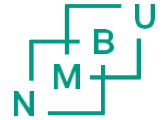
$$\mathbf{a}^{(h)} = \phi(\mathbf{z}^{(h)})$$

(1 x d)      (1 x d)

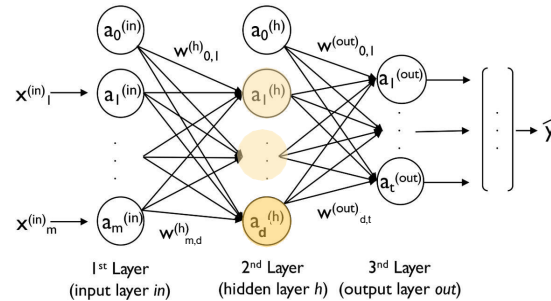
Computations  
for **one sample**  
(row)  $x_i$  in  $X$   
for **all neurons**  
in hidden layer



# A 3-4-3 multi-layer perceptron



$(n \times m)$



$m$ : features  
 $d$ : neurons  
 $\#$  Samples:  $n$

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

$(n \times d)$

$(n \times m)$

$(m \times d)$

$$\mathbf{A}^{(h)} = \phi(\mathbf{Z}^{(h)})$$

$(n \times d)$

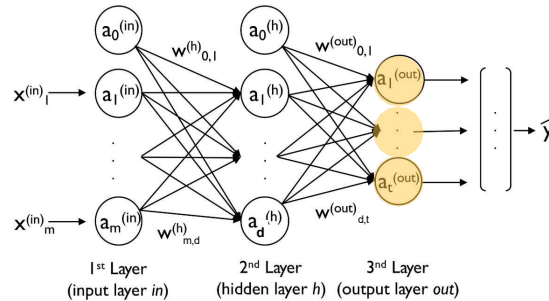
$(n \times d)$

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

# A 3-4-3 multi-layer perceptron



$(n \times m)$



t: labels  
d: neurons  
# Samples: n

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

$(n \times t)$

$(n \times d)$

$(d \times t)$

$$\mathbf{A}^{(out)} = \phi(\mathbf{Z}^{(out)})$$

$(n \times t)$

$(n \times t)$

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