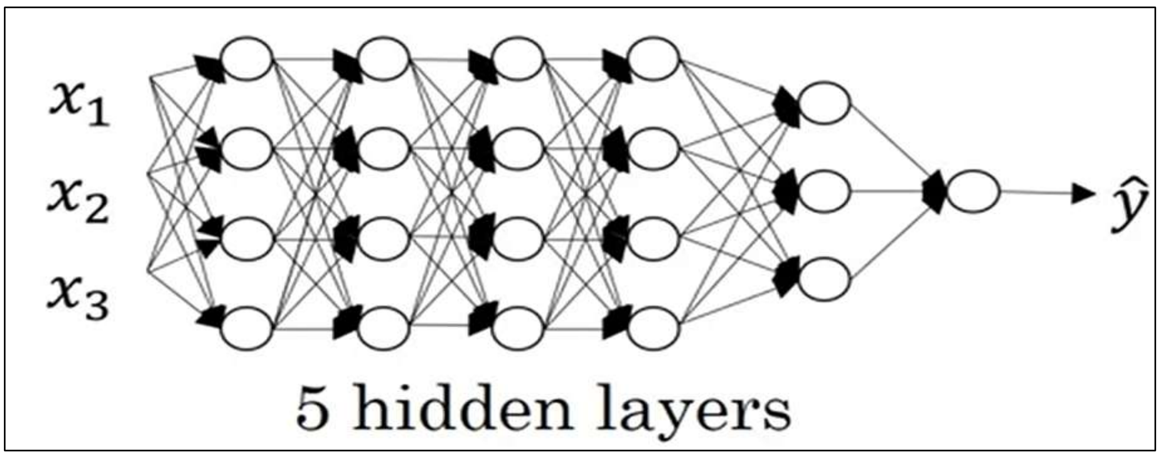
Deep Learning-III

(Deep Neural Networks)

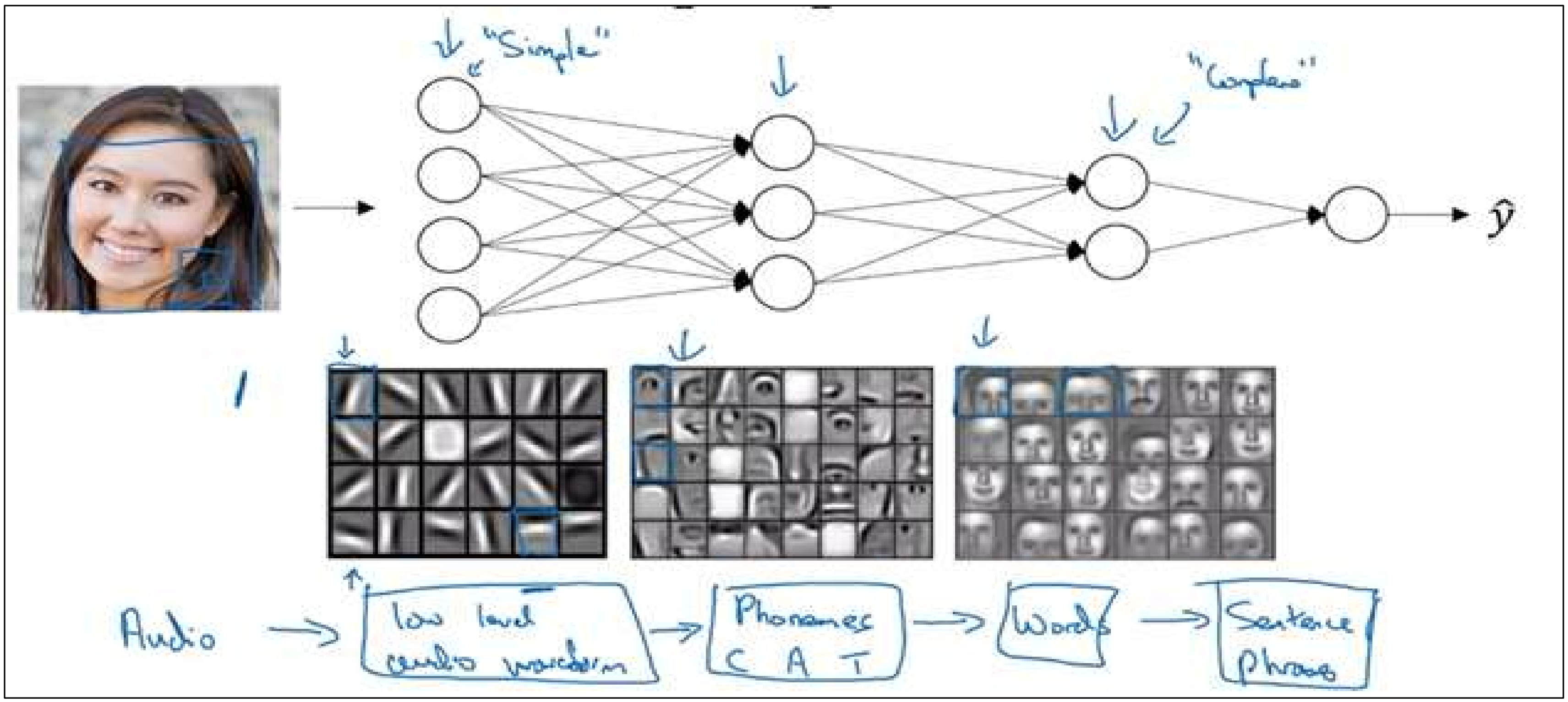
DR. JASMEET SINGH
ASSISTANT PROFESSOR,
CSED, TIET

Deep Neural Networks

- A deep neural network (DNN) is **an artificial neural network (ANN) with multiple hidden layers between the input and output layers**.
- A deep neural network can learn more complex functions, which shallow neural networks fails to learn.



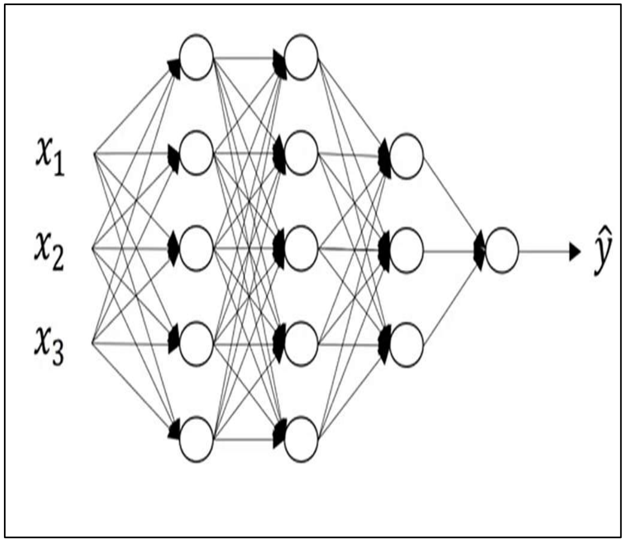
Why Deep Neural Networks?



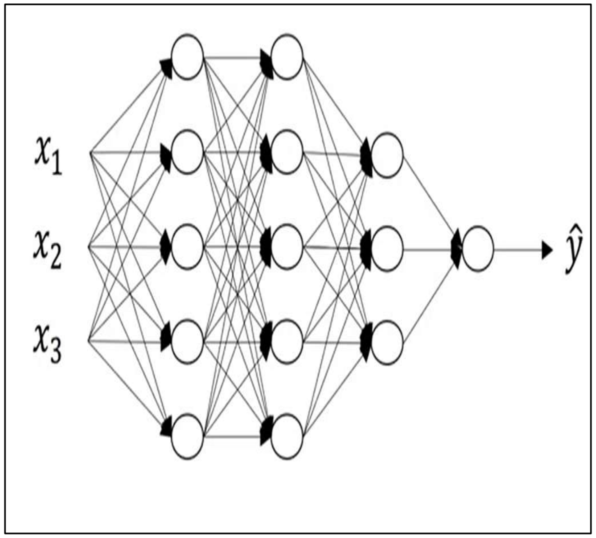
Notations for Deep Neural Networks

- Consider a deep neural network with four layers and 5, 5, 3, and 1 neurons in each layer. Following notations are used for the network:
- L- layer number (L=0 to 4 in this case)
- n_L- number of neurons in each layer (In this case, $n_0=3$, $n_1=n_2=5$, $n_3=3$, $n_4=1$)
- a_L- activations in layer L- produced by applying activation function on the weighted sum of inputs i.e. $a_L=g_L(z_L)$
- W_L- weight matrix at layer L
- B_L- bias matrix at layer L

Weights and bias are used to compute z values of the layer as follows: $zL=W_L a_{L-1}+b_L$



Matrix Dimensions



For one example,

$W_L=(n_L,n_{L-1})$

$b_L=(n_L,1)$

$z_L=a_L=(n_L,1)$

For example, for the neural network shown in fig:

$W_1=(5,3)$, $W_2=(5,5)$, $W_3=(3,5)$, $W_4=(1,3)$

$b_1=(5,1)$, $b_2=(5,1)$, $b_3=(3,1)$, $b_4=(1,1)$

$z_1=(5,1)$, $z_2=(5,1)$, $z_3=(3,1)$, $z_4=(1,1)$

$a_0=(3,1)$, $a_1=(5,1)$, $a_2=(5,2)$, $a_3=(3,1)$, $a_4=(1,1)$

Matrix Dimensions (Contd...)

For n examples/Vectorized implementation,

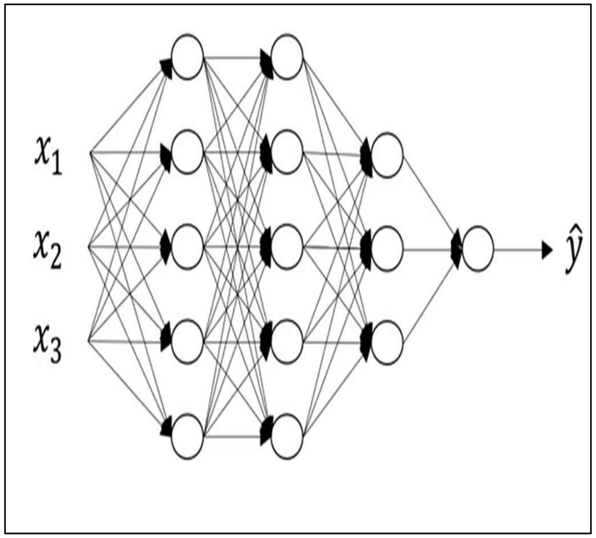
The dimensions of Weights and bias would remain same:

$W_L=(n_L,n_{L-1})$

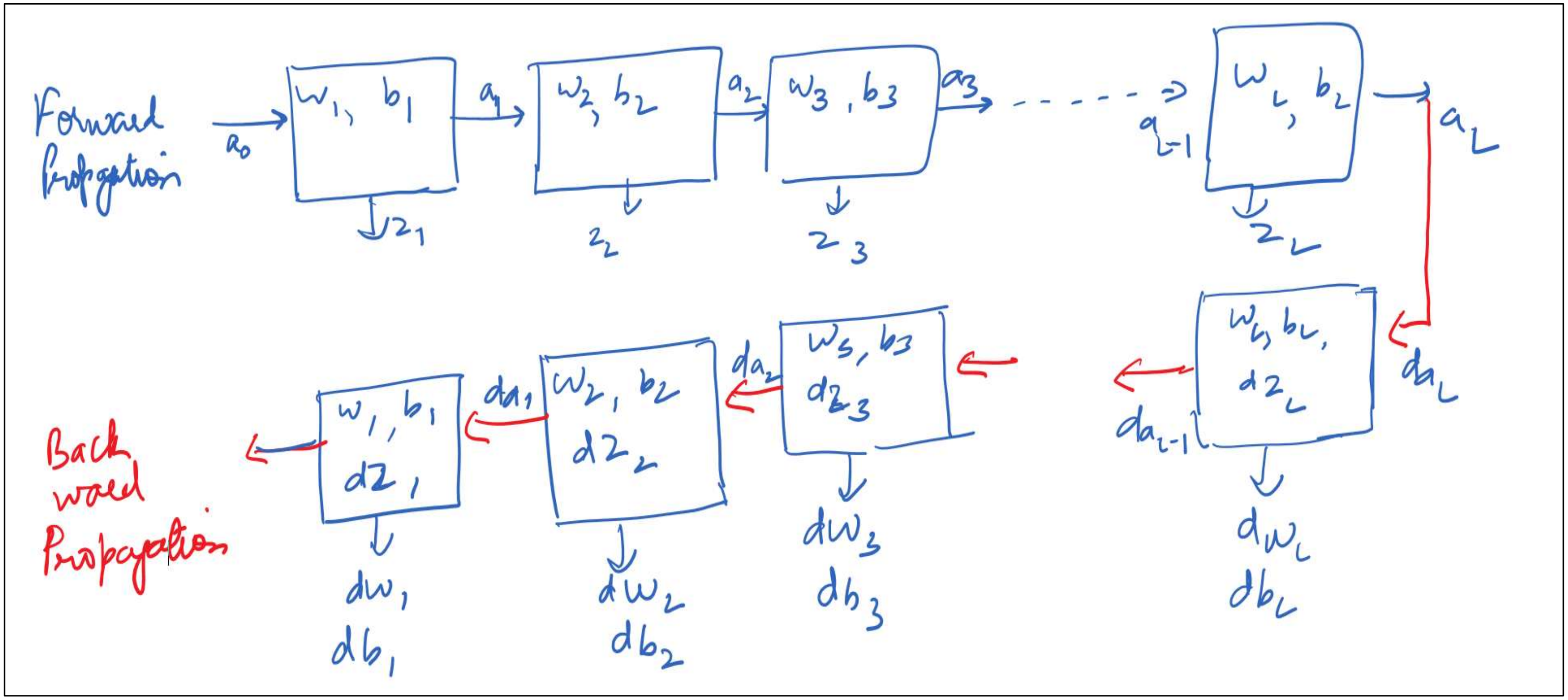
$b_L=(n_L,1)$

The dimensions of Z_L and A_L will be updated according to n examples.

$Z_L=A_L=(n_L,n)$



Forward and Backpropagation Functions



Forward Propagation in a deep network

For one example,

First Layer: $z_1=W_1a_0+b_1$; $a_1=g_1(z_1)$

Second Layer: $z_2=W_2a_1+b_2$; $a_2=g_2(z_2)$

Third Layer: $z_3=W_3a_2+b_3$; $a_3=g_3(z_3)$

Fourth Layer: $z_4=W_4a_3+b_4$; $a_4=g_4(z_4)$

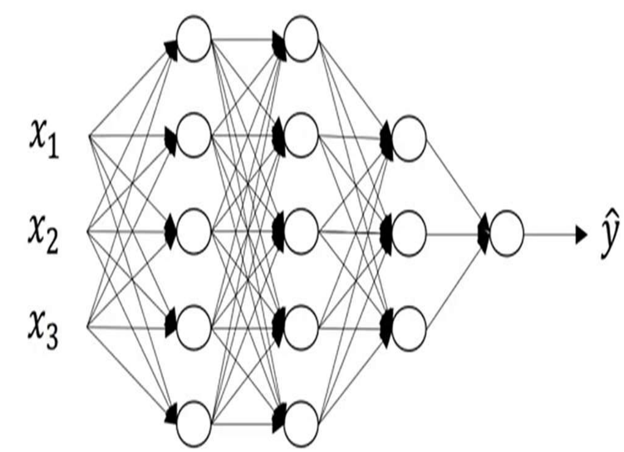
For n training examples,

First Layer: $Z_1=W_1A_0+b_1$; $A_1=g_1(Z_1)$

Second Layer: $Z_2=W_2A_1+b_2$; $A_2=g_2(Z_2)$

Third Layer: $Z_3=W_3A_2+b_3$; $A_3=g_3(Z_3)$

Fourth Layer: $y\hat{}=Z_4=W_4A_3+b_4$; $A_4=g_4(Z_4)$



In general forward propagation can be implemented as:

for l=1 to L:

$Z_l=W_lA_{l-1}+b_1$

$A_l=g_l(Z_l)$

Backward Propagation at Layer L

- One training example

$dz_L = da_L * g_L'(z_L)$

$dW_L = dz_L . a_{L-1}^T$

$db_L = dz_L$

$da_{L-1} = W_L^T . dz_L$

- For n training examples:

$dZ_L = dA_L * g_L'(Z_L)$

$dW_L = \frac{1}{n} dZ_L . A_{L-1}^T$

$db_L = \frac{1}{n} np.sum(dZL, axis = 1, keepdims = True)$

$dA_{L-1} = W_L^T . dZ_L$