Copyright Notice

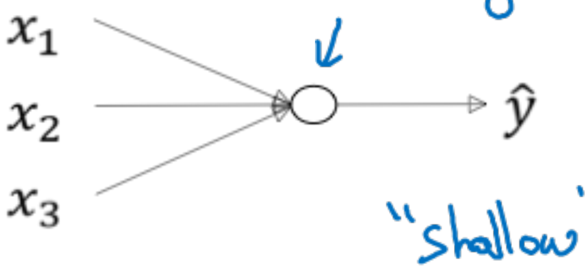
deeplearning.ai

Deep Neural Networks

Deep L-layer Neural network

What is a deep neural network?

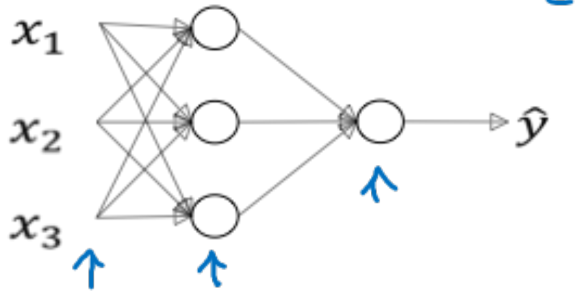
"1 layer NN"



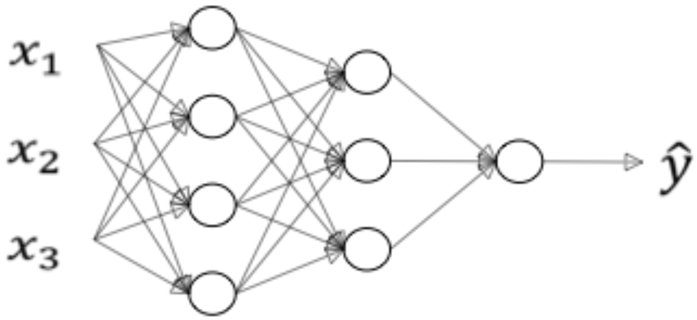
"shallow"

logistic regression

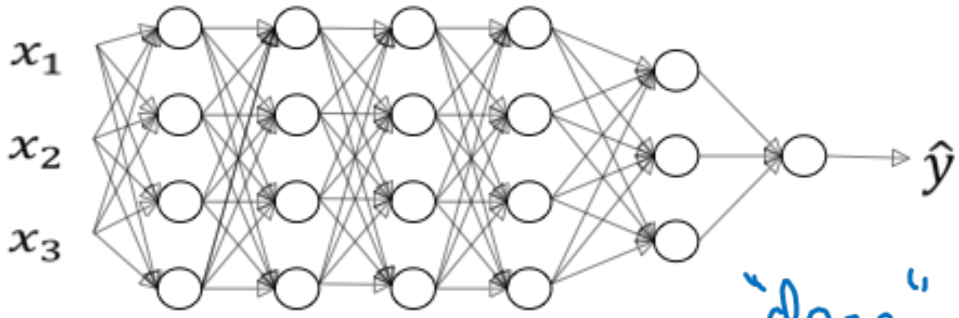
"2 layer" NN



1 hidden layer



2 hidden layers



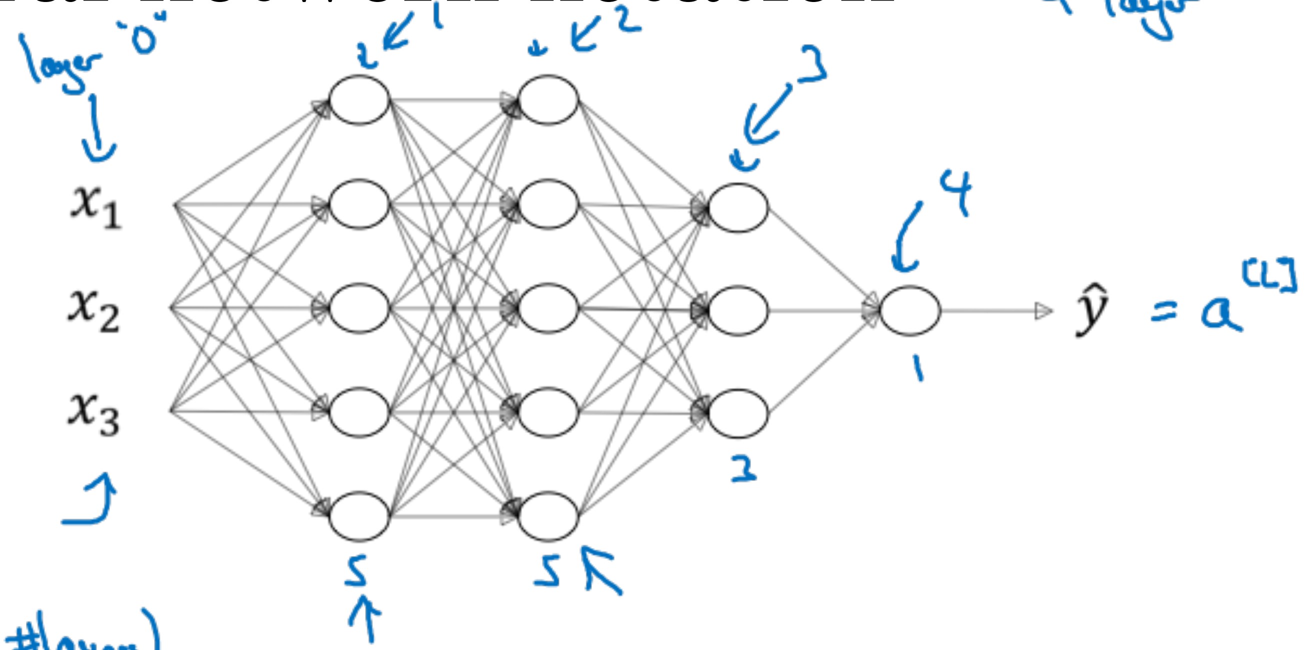
"deep"

5 hidden layers

Deep neural network notation

4 layer NN

$X = a^{[0]}$



$L = 4$ (#layers)

$n^{[l]}$ = #units in layer l

$n^{[1]} = 5,\ n^{[2]} = 5,\ n^{[3]} = 3,\ n^{[4]} = n^{[L]} = 1$

$n^{[0]} = n_x = 3$

$a^{[l]}$ = activations in layer l

$a^{[l]} = g^{[l]}(z^{[l]})$, $W^{[l]}$ = weights for $\underline{z^{[l]}}$

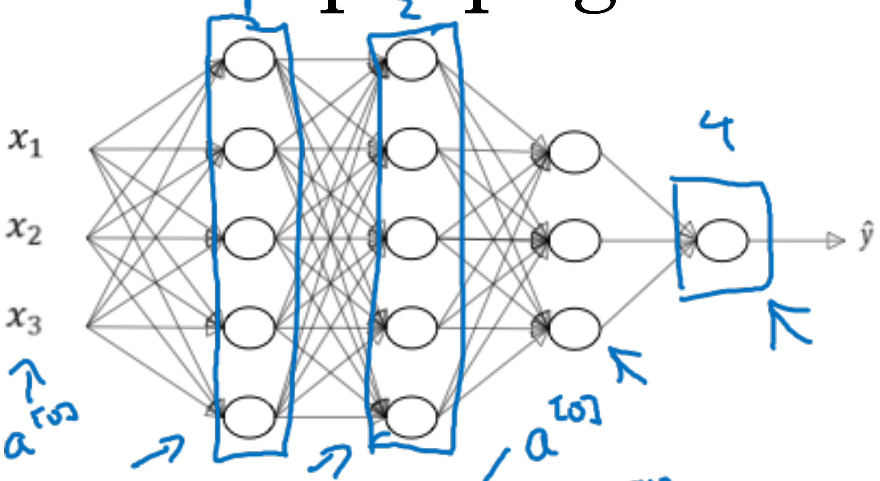
$b^{[l]}$

deeplearning.ai

Deep Neural Networks

Forward Propagation in a Deep Network

Forward propagation in a deep network



X : $z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$

$a^{[1]} = g^{[1]}(z^{[1]})$

$z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$

$a^{[2]} = g^{[2]}(z^{[2]})$

$z^{[4]} = W^{[4]} a^{[2]} + b^{[4]}$, $a^{[4]} = g^{[4]}(z^{[4]})$
$= \hat{y}$

$\left[z^{[1](1)}, z^{[1](2)}, \dots z^{[1](m)} \right]$

$A^{[0]} = X$

$$\boxed{\begin{aligned} Z^{[l]} &= W^{[l]} A^{[l-1]} + b^{[l]} \\ A^{[l]} &= g^{[l]}(Z^{[l]}) \end{aligned}}$$

Vectorized:

$Z^{[1]} = W^{[1]} \cancel{X} A^{[0]} + b^{[1]}$

$A^{[1]} = g^{[1]}(Z^{[1]})$

$Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$

$A^{[2]} = g^{[2]}(Z^{[2]})$

$\hat{Y} = g(Z^{[4]}) = A^{[4]}$

→ $X = A^{[0]}$

for $l = 1..4$

Deep Neural Networks

Getting your matrix dimensions right

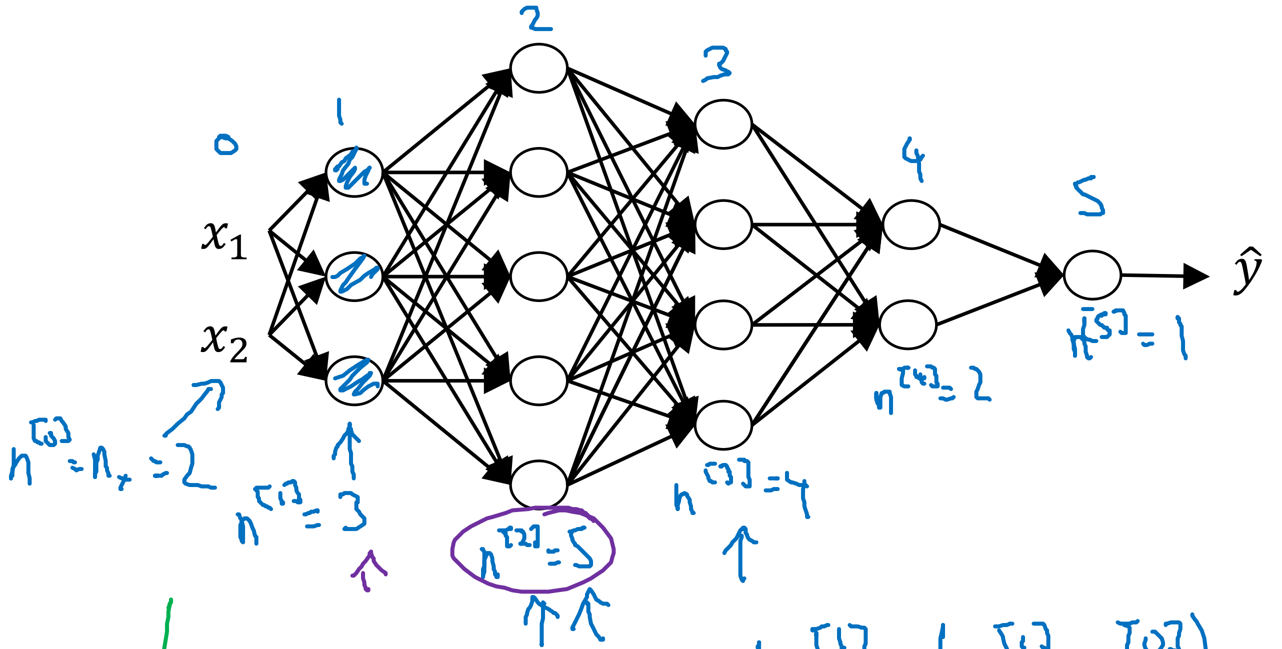
deeplearning.ai

□□□□□ □t□□□□$W^{[l]}$ □□□□$b^{[l]}$

$L=5$

$z^{[l]} = g^{[l]}(a^{[l]})$

$a^{[l]}$



$W^{[l]} : (n^{[l]}, n^{[l-1]})$

$b^{[l]} : (n^{[l]}, 1)$

$dW^{[l]} : (n^{[l]}, n^{[l-1]})$

$db^{[l]} : (n^{[l]}, 1)$

$z^{[1]} = \boxed{W^{[1]} \cdot x} + \boxed{b^{[1]}}$

$(3,1) \leftarrow (3,2) \quad (2,1) \quad (3,1)$

$(n^{[1]},1) \quad (n^{[1]},n^{[0]}) \quad (n^{[0]},1) \quad (n^{[1]},1)$

$\begin{bmatrix}\cdot\\\cdot\\\cdot\end{bmatrix} = \begin{bmatrix}\cdot&\cdot\\\cdot&\cdot\\\cdot&\cdot\end{bmatrix} \begin{bmatrix}\cdot\\\cdot\end{bmatrix}$

$W^{[1]} : (n^{[1]}, n^{[0]})$

$W^{[2]} : (5,3) \quad (n^{[2]}, n^{[1]})$

$z^{[2]} = \boxed{W^{[2]} \cdot a^{[1]}} + \boxed{b^{[2]}}$

$\rightarrow (5,1) \quad (5,3) \quad (3,1) \quad (5,1)$

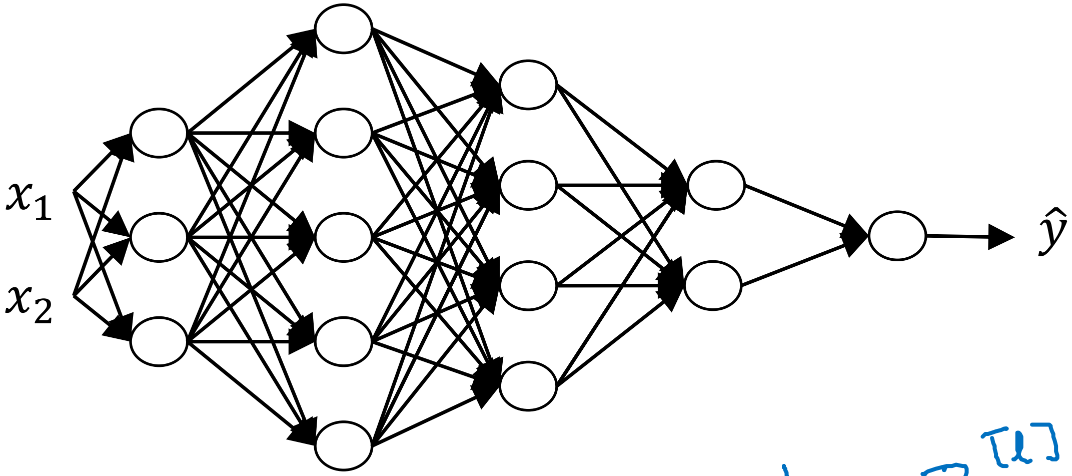
$(n^{[2]},1)$

$W^{[3]} : (4,5)$

$W^{[4]} : (2,4) \quad , \quad W^{[5]} : (1,2)$

Andrew Ng

□ □□□□□Z□□□□□ □□□ □□□□□□□



$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$

$(n^{[1]},1)$ $(n^{[1]},n^{[0]})$ $\underline{(n^{[0]},1)}$ $(n^{[1]},1)$

$$\left[z^{[1](1)} \; z^{[1](2)} \cdots z^{[1](m)} \right]$$

$$Z^{[1]} = W^{[1]} \cdot X + b^{[1]}$$

$\underline{(n^{[1]},m)}$ $\underbrace{\underline{(n^{[1]},n^{[0]})} \quad \underline{(n^{[0]},m)}}$ $\underline{(n^{[1]},1)}$

$(n^{[1]},m)$

$$z^{[l]}, a^{[l]} : (n^{[l]},1)$$

$$Z^{[l]}, A^{[l]} : (n^{[l]},m)$$

$$l=0 \qquad A^{[0]} = X = (n^{[0]},m)$$

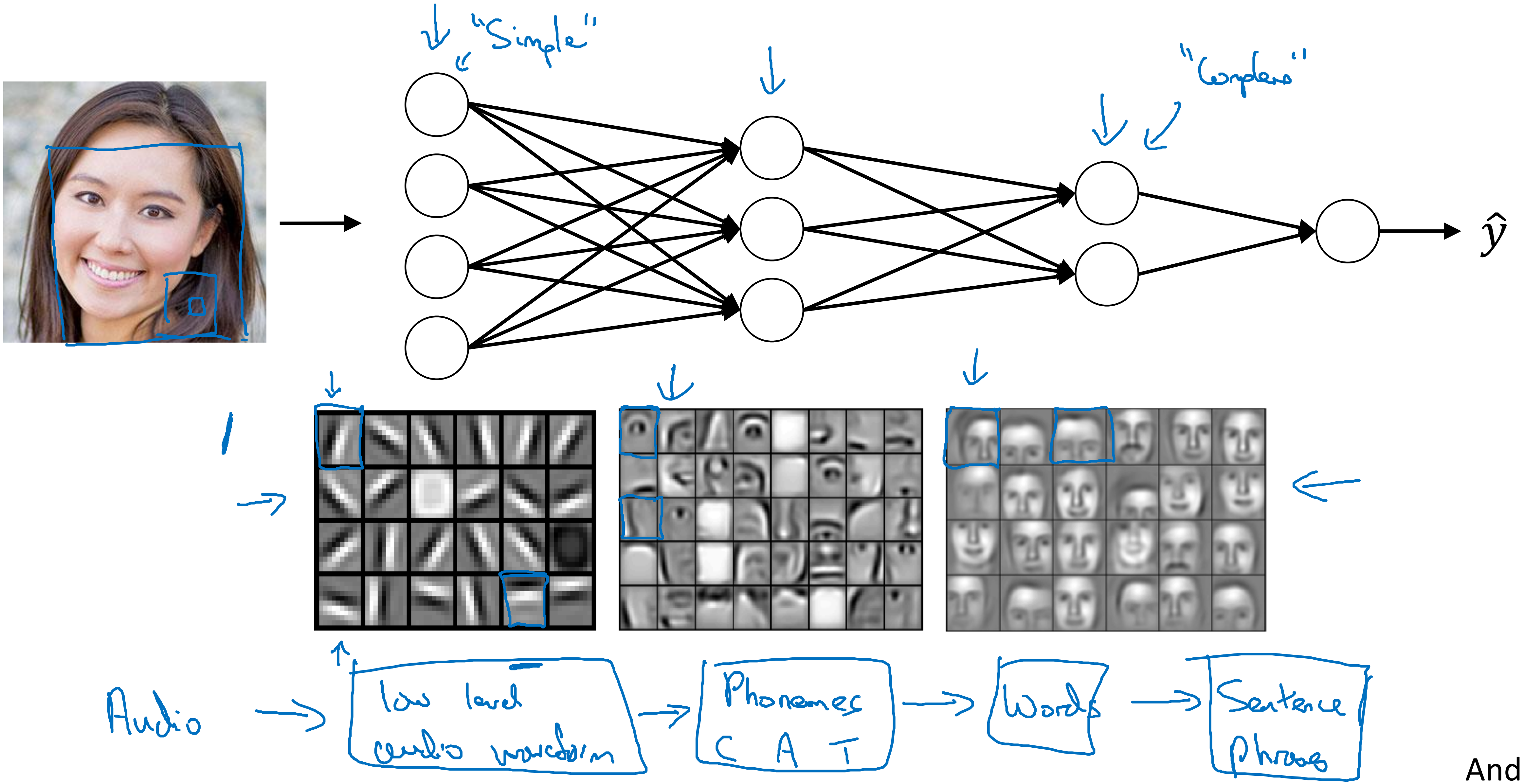
$$dZ^{[l]}, dA^{[l]} : (n^{[l]},m)$$

Andrew Ng

deeplearning.ai

Deep Neural Networks

Why deep representations?

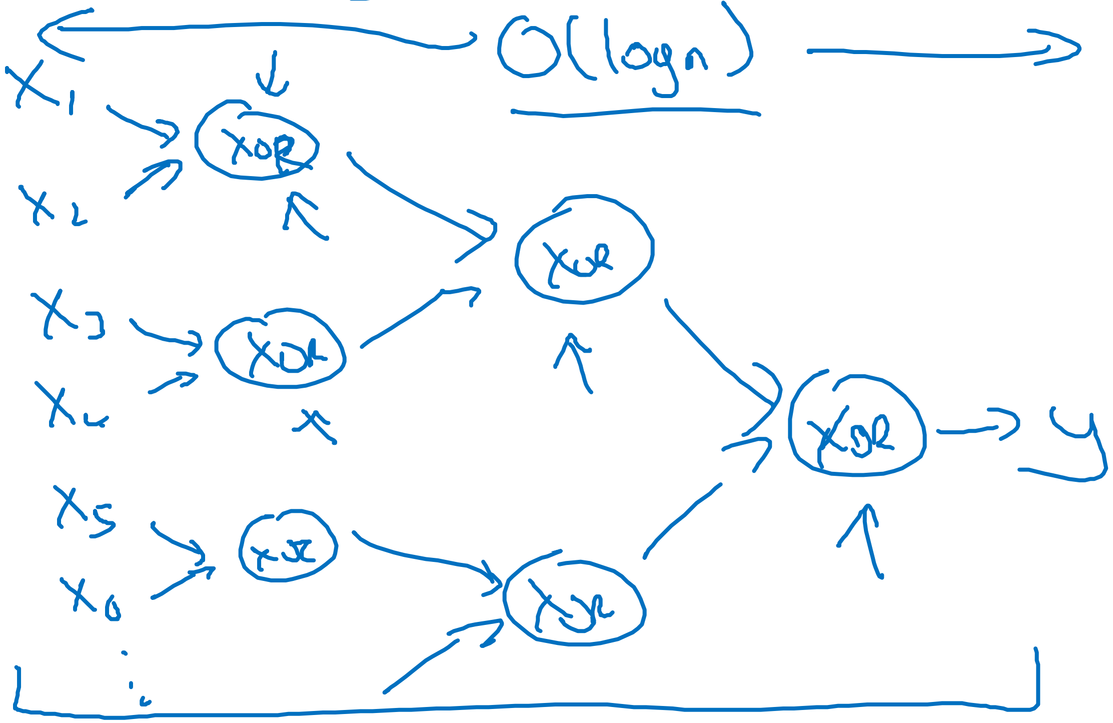


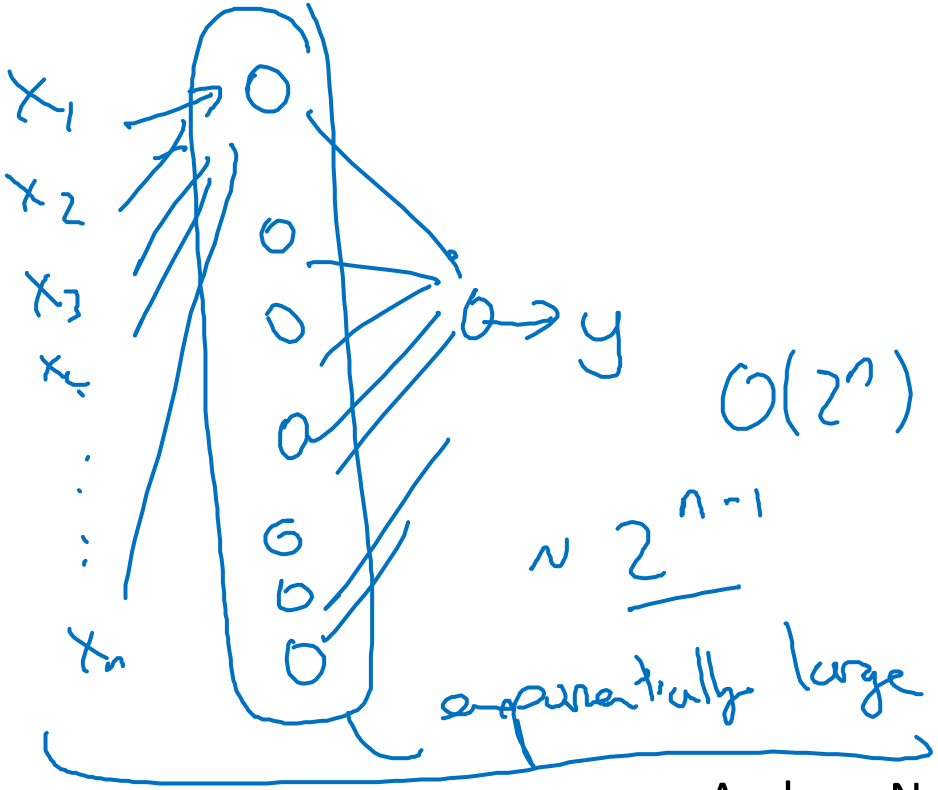
"Simple"
"Complex"
$\hat{y}$
Audio
low level audio waveform
Phonemes
C A T
Words
Sentence phrase

[illegible]

[illegible]

$y = x_1 \text{ XOR } x_2 \text{ XOR } x_3 \text{ XOR} \ldots \text{ XOR } x_n$





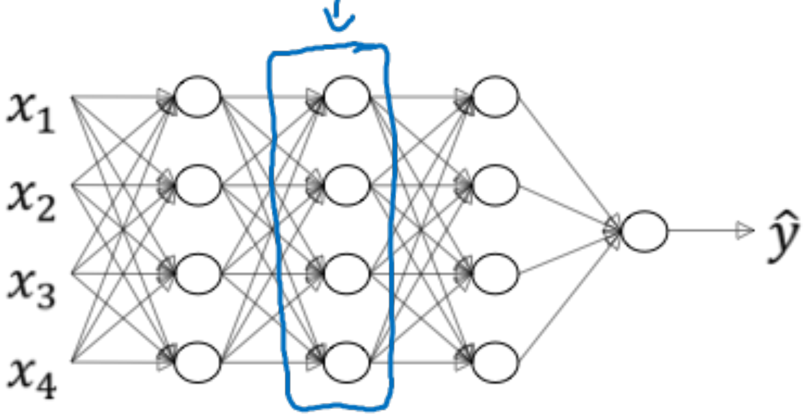
Andrew Ng

deeplearning.ai

Deep Neural Networks

Building blocks of deep neural networks

Forward and backward functions



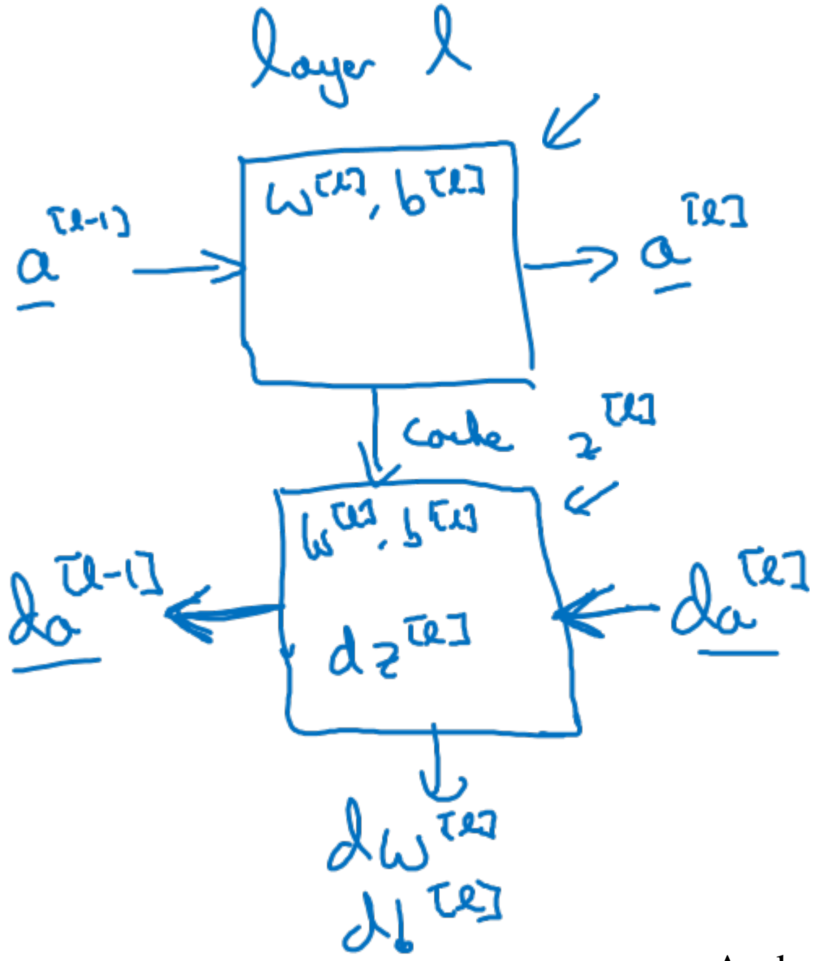
Layer l: $W^{[l]}, b^{[l]}$

→ Forward: Input $a^{[l-1]}$, output $a^{[l]}$

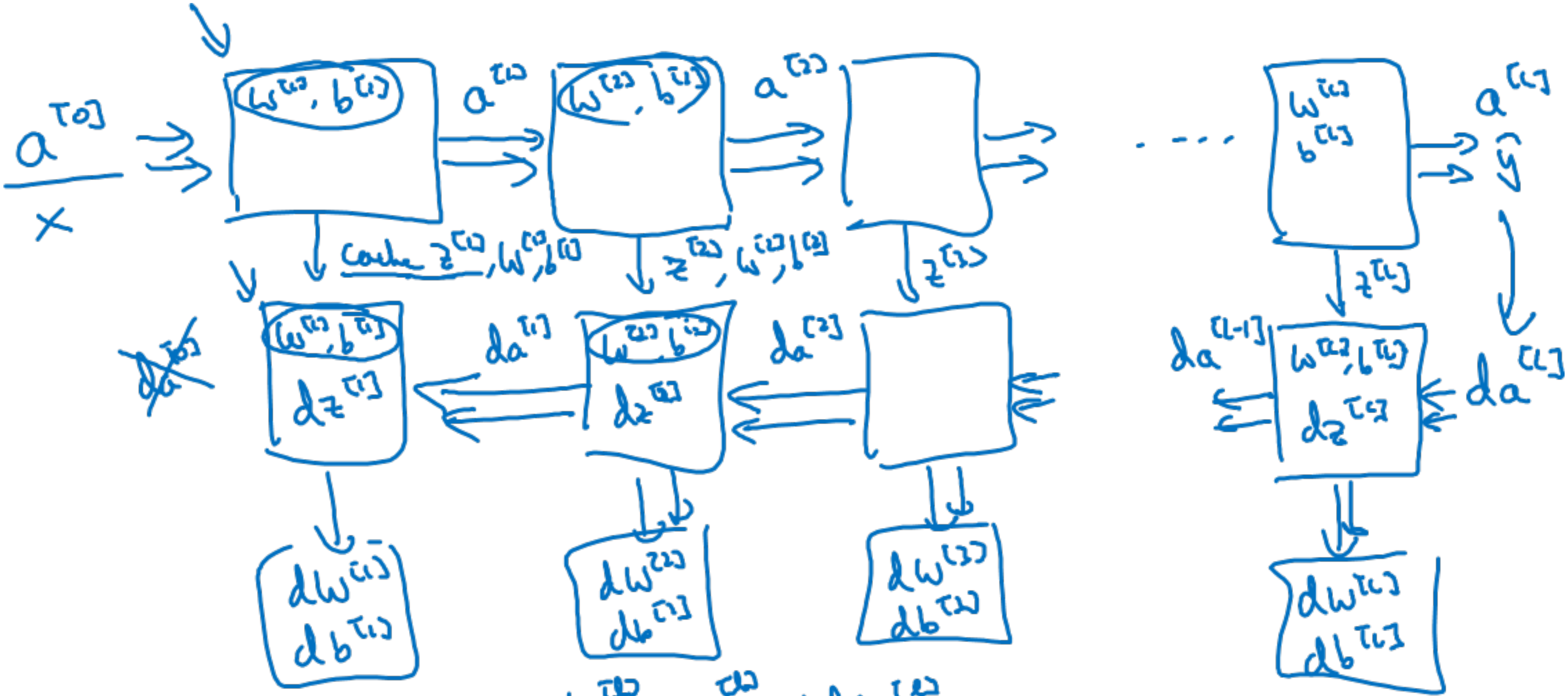
$\underline{z^{[l]}} = W^{[l]} \underline{a^{[l-1]}} + b^{[l]}$ cache $\underline{z^{[l]}}$

$\underline{a^{[l]}} = g^{[l]}(z^{[l]})$

→ Backward: Input $da^{[l]}$, cache$(z^{[l]})$ output $da^{[l-1]}$, $dW^{[l]}$, $db^{[l]}$



Forward and backward functions



$$W^{[l]} := W^{[l]} - \alpha dW^{[l]}$$

$$b^{[l]} := b^{[l]} - \alpha db^{[l]}$$

deeplearning.ai

Deep Neural Networks

Forward and backward propagation

Backward propagation for layer l

→ Input $da^{[l]}$

→ Output $da^{[l-1]}, dW^{[l]}, db^{[l]}$

$$\underline{dz^{[l]}} = \underbrace{da^{[l]}}_{} * g^{[l]'}(z^{[l]})$$

$$dW^{[l]} = dz^{[l]} \cdot \underline{a^{[l-1]}}$$

$$db^{[l]} = dz^{[l]}$$

$$\underbrace{da^{[l-1]}}_{} = W^{[l]T} \cdot dz^{[l]}$$

$$dz^{[l]} = W^{[l+1]T} dz^{[l+1]} * g^{[l]'}(z^{[l]})$$

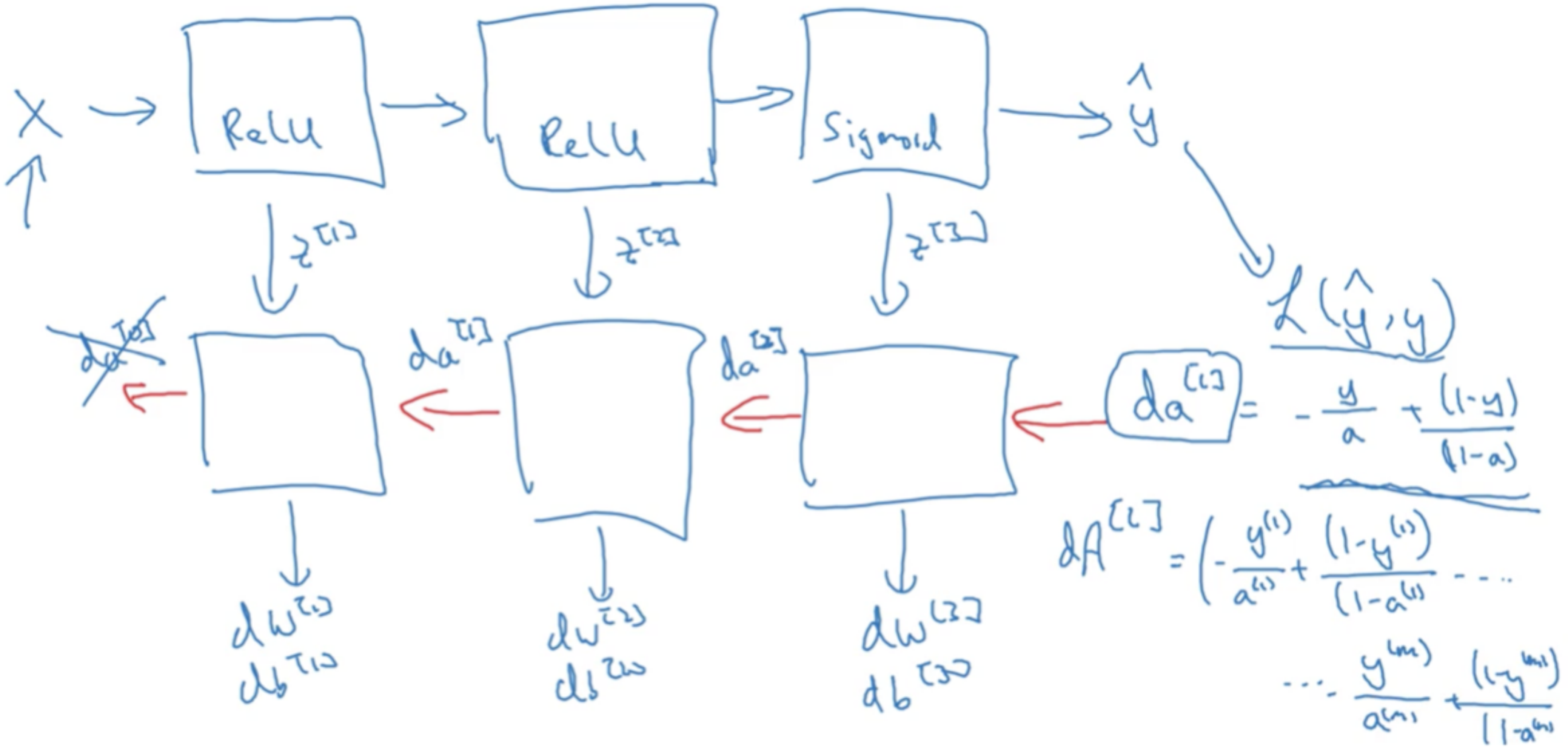
$$dZ^{[l]} = dA^{[l]} * g^{[l]'}(Z^{[l]})$$

$$dW^{[l]} = \frac{1}{m} dZ^{[l]} \cdot A^{[l-1]T}$$

$$db^{[l]} = \frac{1}{m} \text{np.sum}(dZ^{[l]}, \text{axis}=1, \text{keepdims}=\text{True})$$

$$dA^{[l-1]} = W^{[l]T} \cdot dZ^{[l]}$$

Summary


X
ReLU
ReLU
Sigmoid
$\hat{y}$
$z^{[1]}$
$z^{[2]}$
$z^{[3]}$
$\mathcal{L}(\hat{y}, y)$
~~$da^{[0]}$~~
$da^{[1]}$
$da^{[2]}$
$da^{[l]} = -\frac{y}{a} + \frac{(1-y)}{(1-a)}$
$dA^{[l]} = \left(-\frac{y^{(1)}}{a^{(1)}} + \frac{(1-y^{(1)})}{(1-a^{(1)})} \cdots \right.$
$\cdots -\frac{y^{(m)}}{a^{(m)}} + \frac{(1-y^{(m)})}{(1-a^{(m)})}$
$dW^{[1]}$
$db^{[1]}$
$dW^{[2]}$
$db^{[2]}$
$dW^{[3]}$
$db^{[3]}$

deeplearning.ai

Deep Neural Networks

Parameters vs Hyperparameters

□ □□□□□□□□□y□□□□□□□□ □□□□□□

□□□□□ □t□□□□□ $\underline{W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]}}$ …

Hyperparameters: learning rate $\underline{\alpha}$

#iterations

#hidden layers L

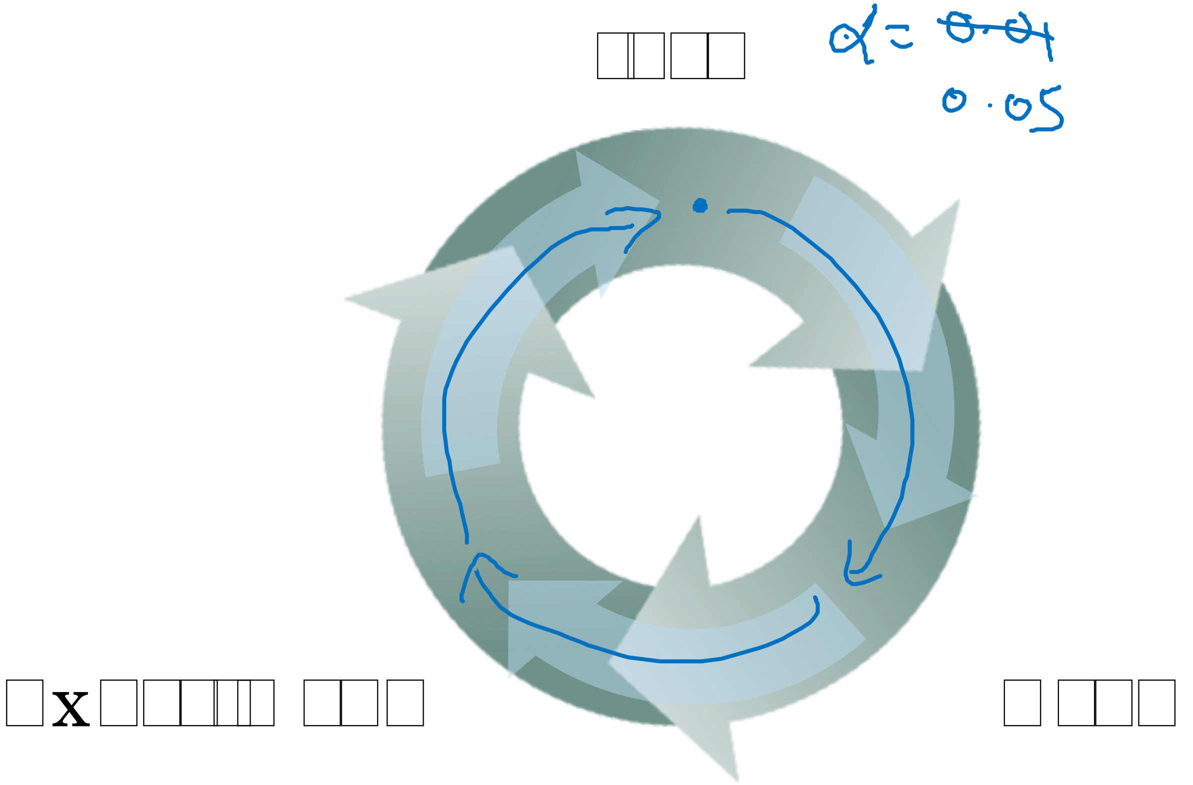
#hidden units $n^{[1]}, n^{[2]}, \ldots$

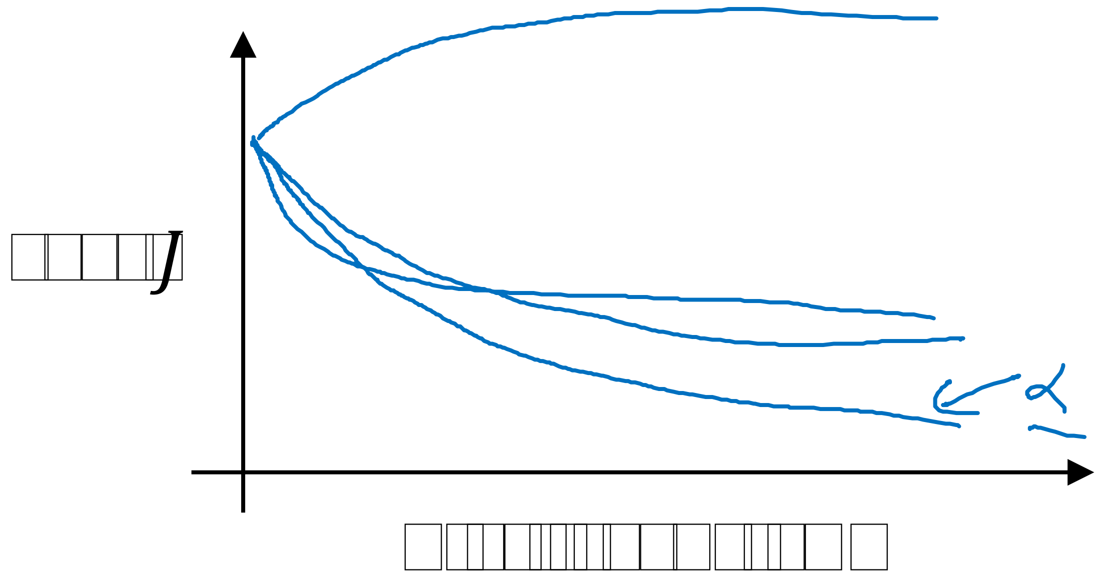
choice of activation function

Later: Momentum, mini-batch size, regularizations, …

Andrew Ng

[illegible] y [illegible]





Vision, Speech, NLP, Ad, Search, Recommendation.

Andrew Ng

$$dZ^{[L]} = A^{[L]} - Y$$

$$dW^{[L]} = \frac{1}{m} dZ^{[L]} A^{[L-1]^T}$$

$$db^{[L]} = \frac{1}{m} np.sum(dZ^{[L]}, axis = 1, keepdims = True)$$

$$dZ^{[L-1]} = W^{[L]^T} dZ^{[L]} * g'^{[L-1]}(Z^{[L-1]})$$

Note that * denotes element-wise multiplication)

$$\vdots$$

$$dZ^{[1]} = W^{[2]^T} dZ^{[2]} * g'^{[1]}(Z^{[1]})$$

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} A^{[0]^T}$$

Note that $A^{[0]^T}$ is another way to denote the input features, which is also written as X^T

$$db^{[1]} = \frac{1}{m} np.sum(dZ^{[1]}, axis = 1, keepdims = True)$$