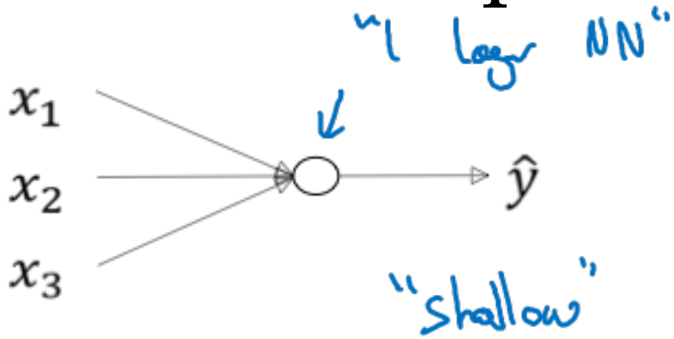
Copyright Notice

Deep Neural Networks

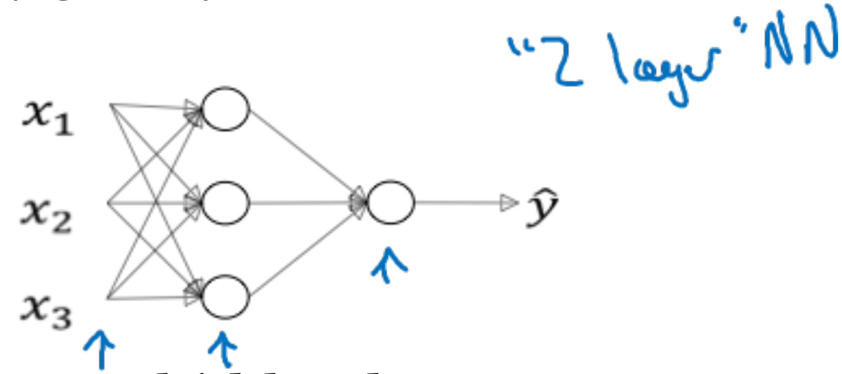
deeplearning.ai

Deep L-layer Neural network

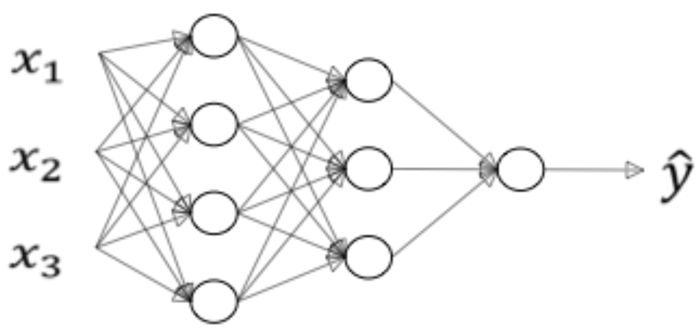
What is a deep neural network?



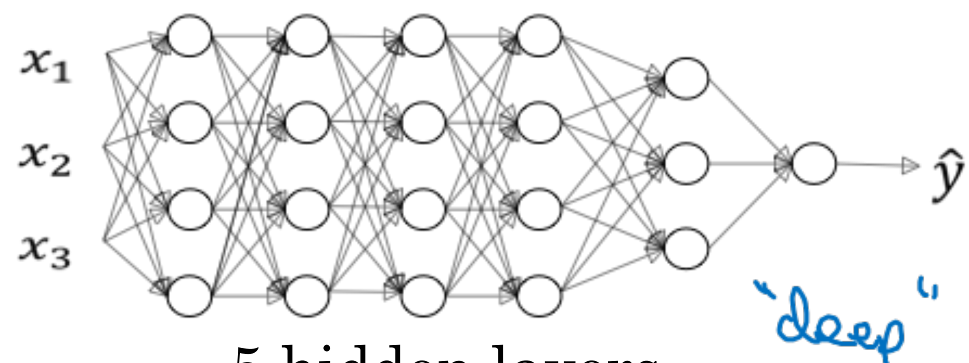
logistic regression



1 hidden layer



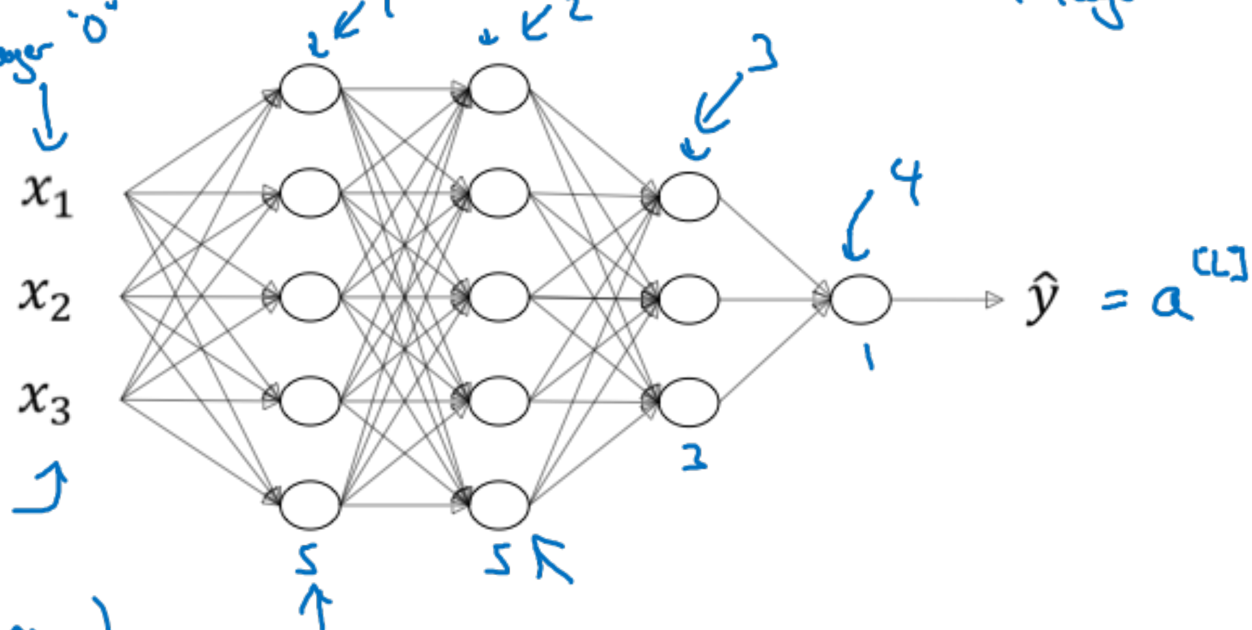
2 hidden layers



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 $\underline{z^{[l]}}$

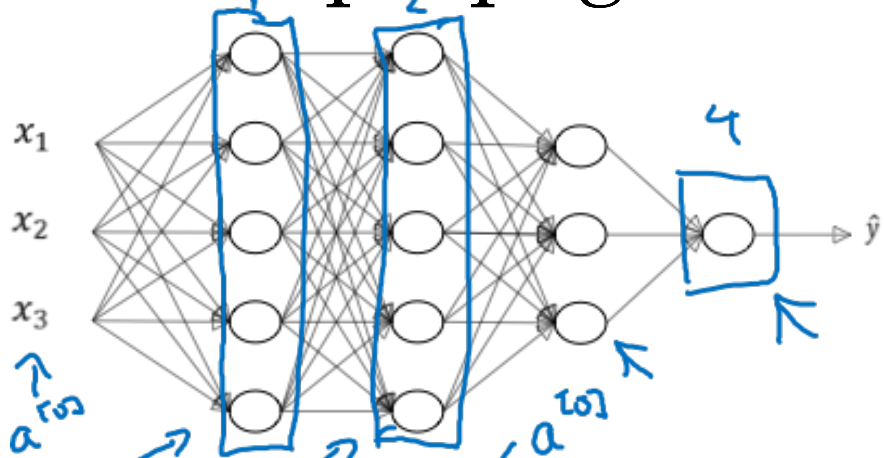
$b^{[l]}$

deeplearning.ai

Deep Neural Networks

Forward Propagation in a Deep Network

Forward propagation in a deep network



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

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

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

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

$\vdots$

$z^{[4]} = W^{[4]} a^{[3]} + b^{[4]}$, $a^{[4]} = g^{[4]}(z^{[4]}) = \hat{y}$

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

$A^{[0]} = X$

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

Vectorized:

$X = A^{[0]}$

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

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

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

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

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

for $l = 1..4$

deeplearning.ai

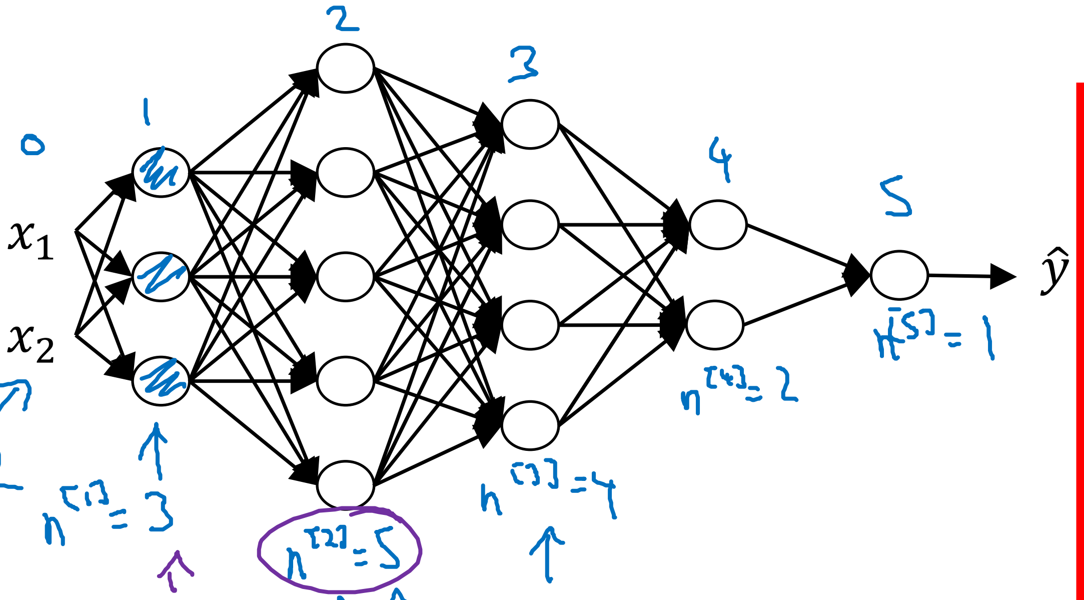
Deep Neural Networks

Getting your matrix dimensions right

Parameters $W^{[l]}$ and $b^{[l]}$

$L = 5$

DIMENSIONS.These will remain same



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

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

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

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

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

$a^{[l]}$

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

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

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

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

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

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

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

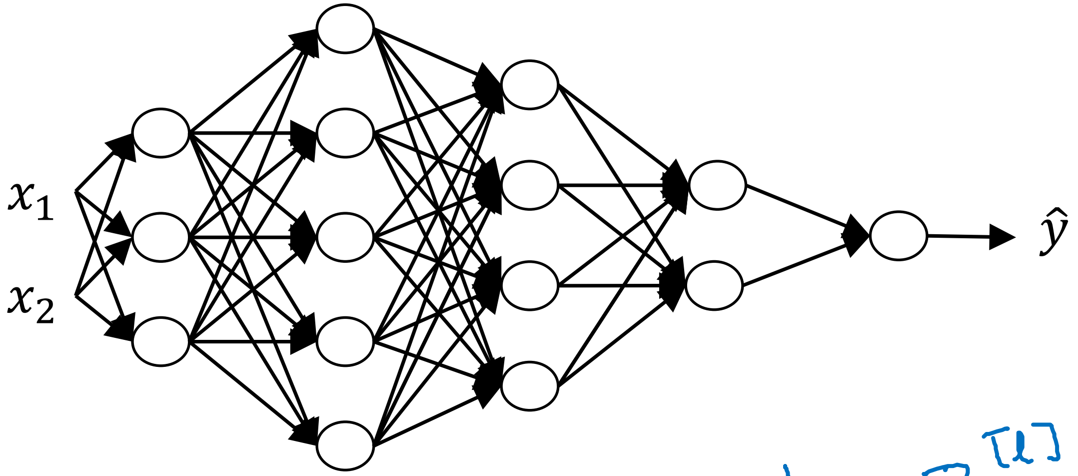
$(n^{[2]},1)$

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

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

Andrew Ng

Vectorized implementation



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

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

$\begin{bmatrix} z^{[1](1)} & z^{[1](2)} & \cdots & z^{[1](m)} \end{bmatrix}$

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

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

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

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

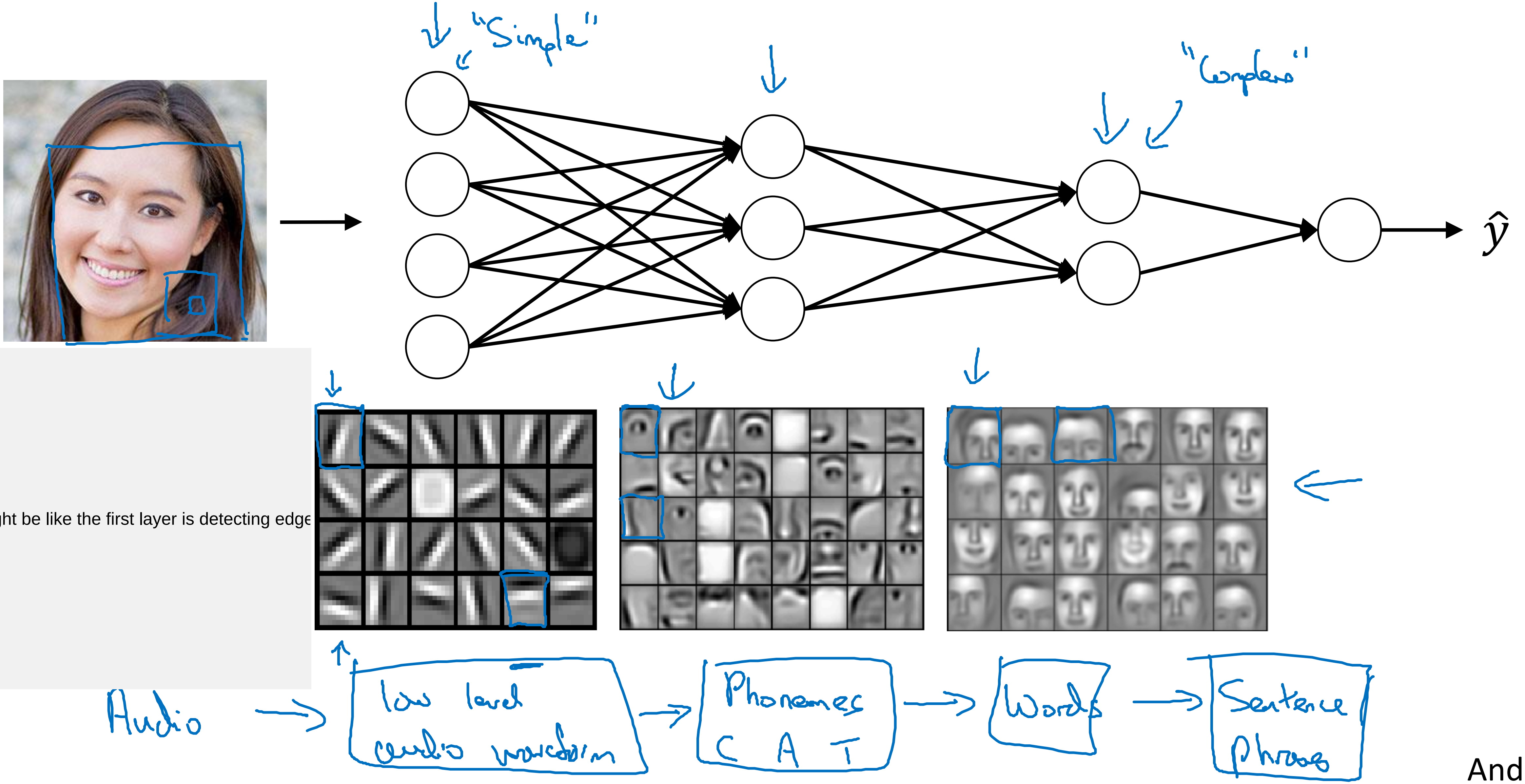
$\boxed{\begin{array}{l} Z^{[l]}, A^{[l]} : (n^{[l]}, m) \\ \quad l = 0 \quad A^{[0]} = X = (n^{[0]}, m) \\ dZ^{[l]}, dA^{[l]} : (n^{[l]}, m) \end{array}}$

Deep Neural Networks

deeplearning.ai

Why deep representations?

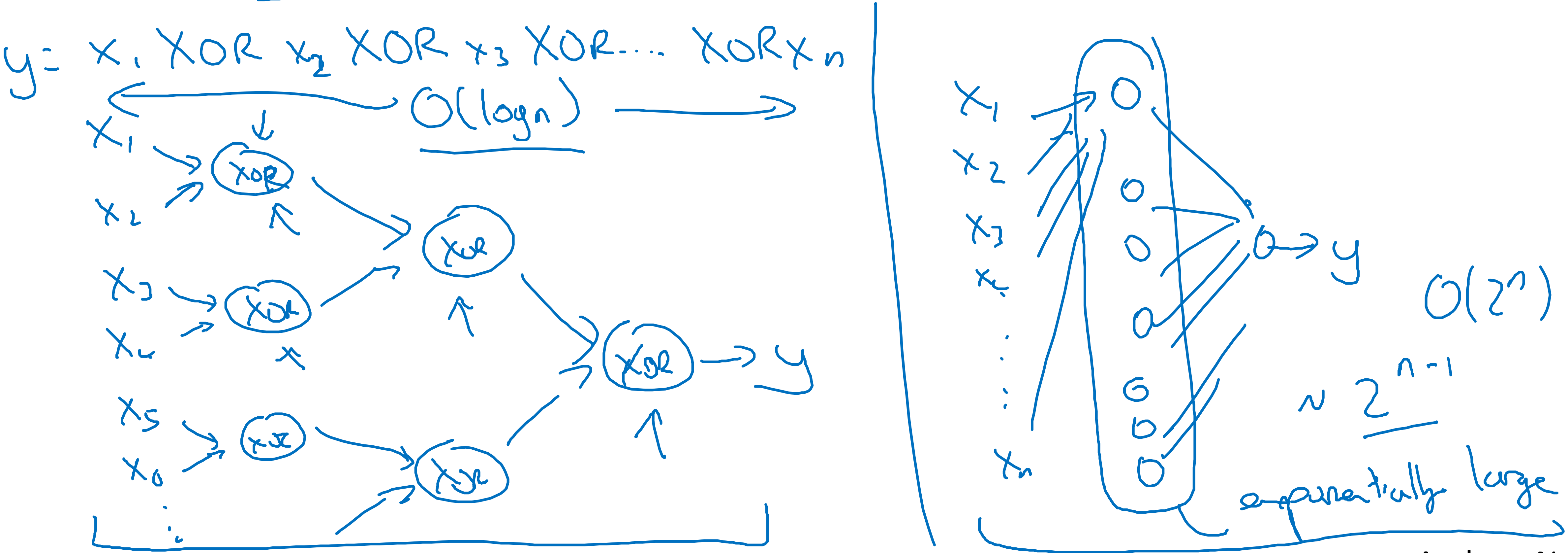
Intuition about deep representation



It might be like the first layer is detecting edge

Circuit theory and deep learning

Informally: There are functions you can compute with a "small" L-layer deep neural network that shallower networks require exponentially more hidden units to compute.



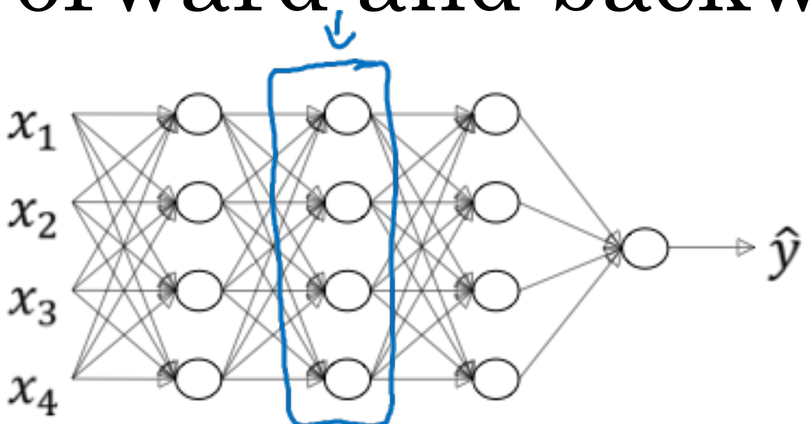
Andrew Ng

Deep Neural Networks

deeplearning.ai

Building blocks of deep neural networks

Forward and backward functions



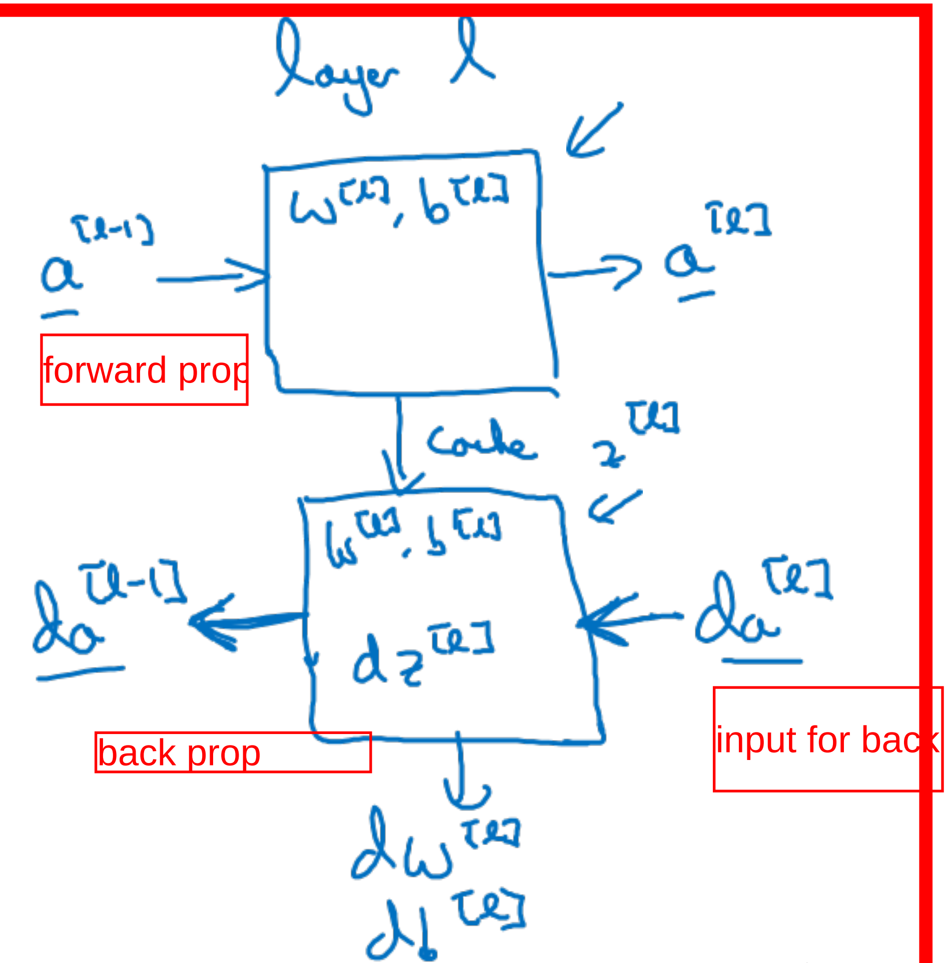
Layer l: $W^{[l]}, b^{[l]}$

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

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

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

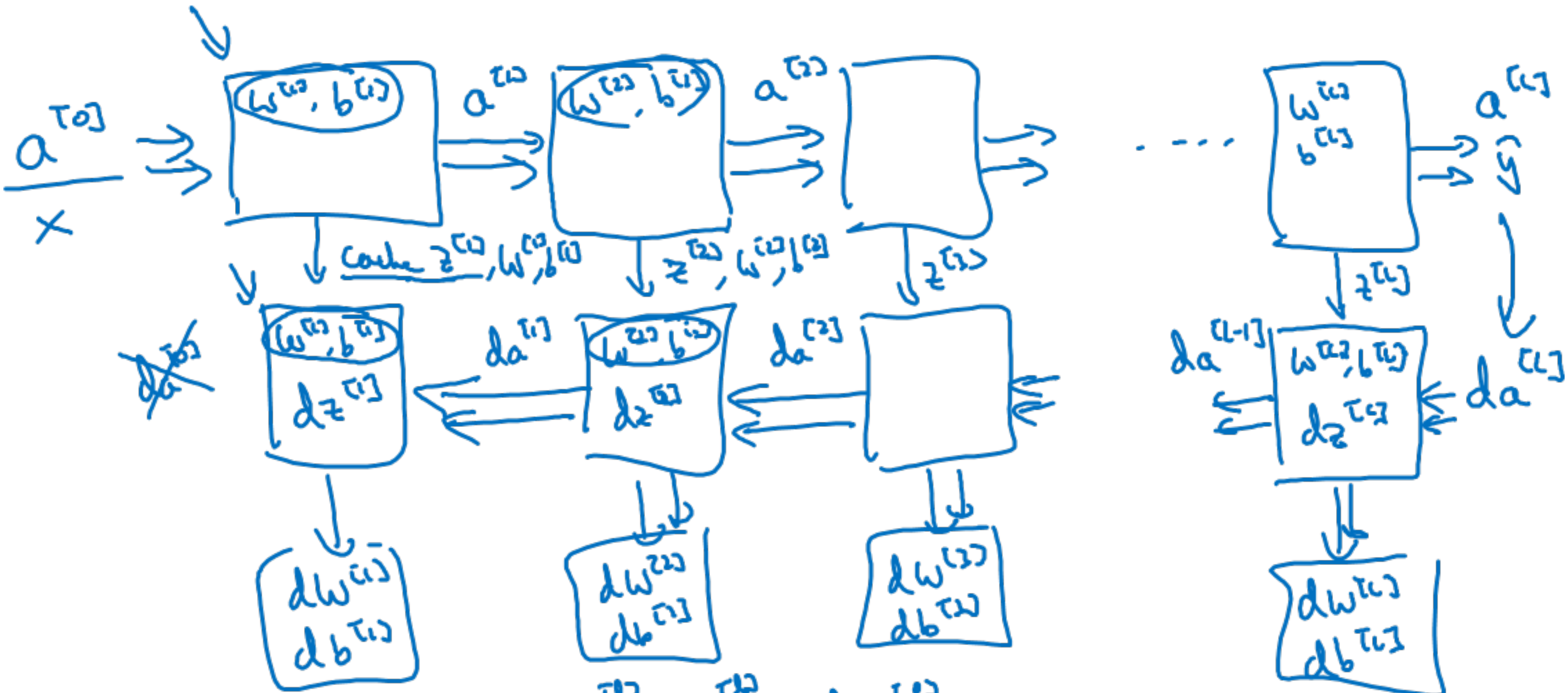
→ Backward: Input $da^{[l]}$, cache($z^{[l]}$) output $da^{[l-1]}$, $dW^{[l]}$, $db^{[l]}$



Andrew N

During forward propogation step we will also cache the Z value for later use in backprop step . We will also cache the W and b

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

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

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

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

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

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

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

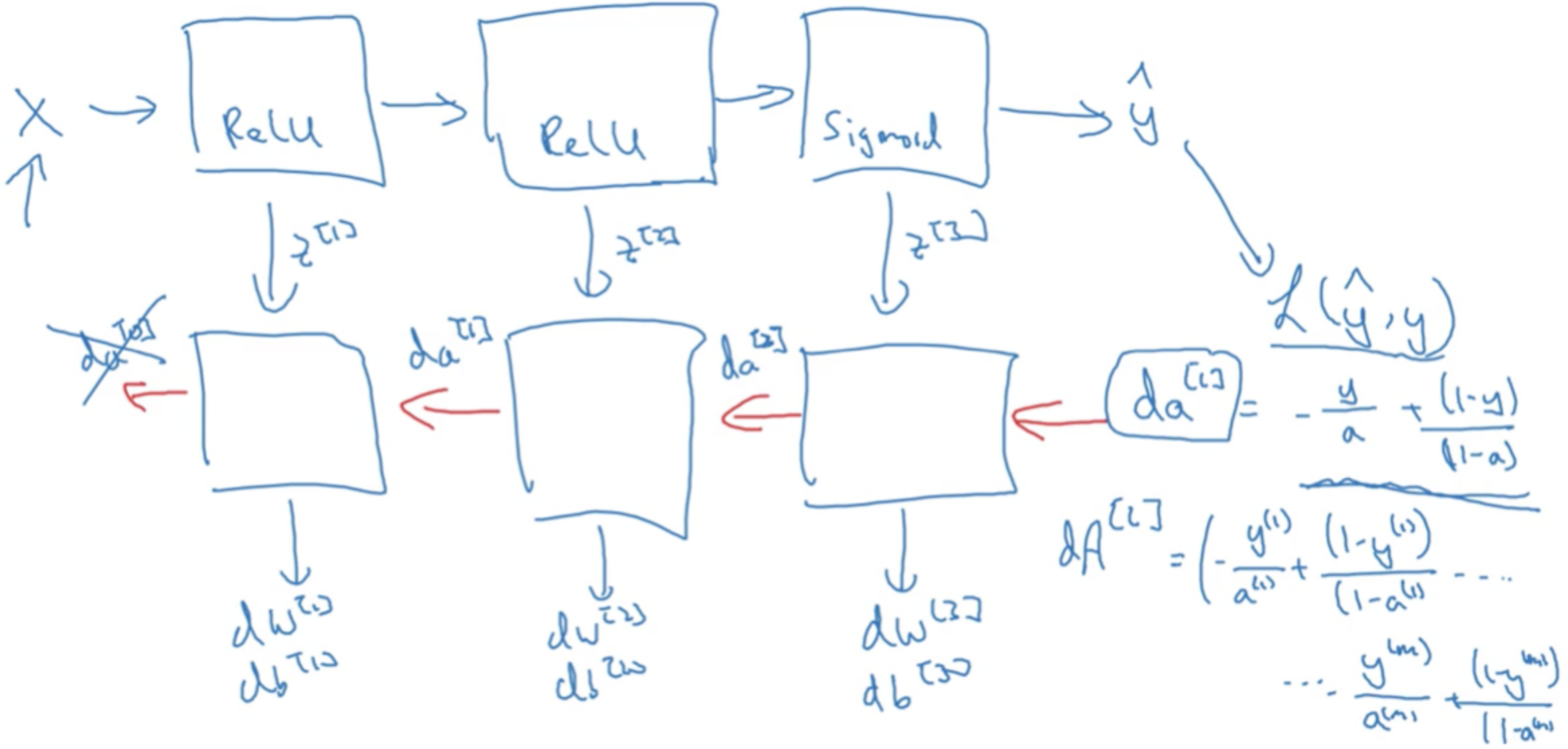
$$dW^{[l]} = \frac{1}{m} dZ^{[l]} \cdot A^{[l-1]T}$$

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

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

Summary

But during backproopi how to initialise the dA[l]. It turn out that ki it c



deeplearning.ai

Deep Neural Networks

Parameters vs Hyperparameters

What are hyperparameters?

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]}$ ...

Hyperparameters: learning rate α

#iterations

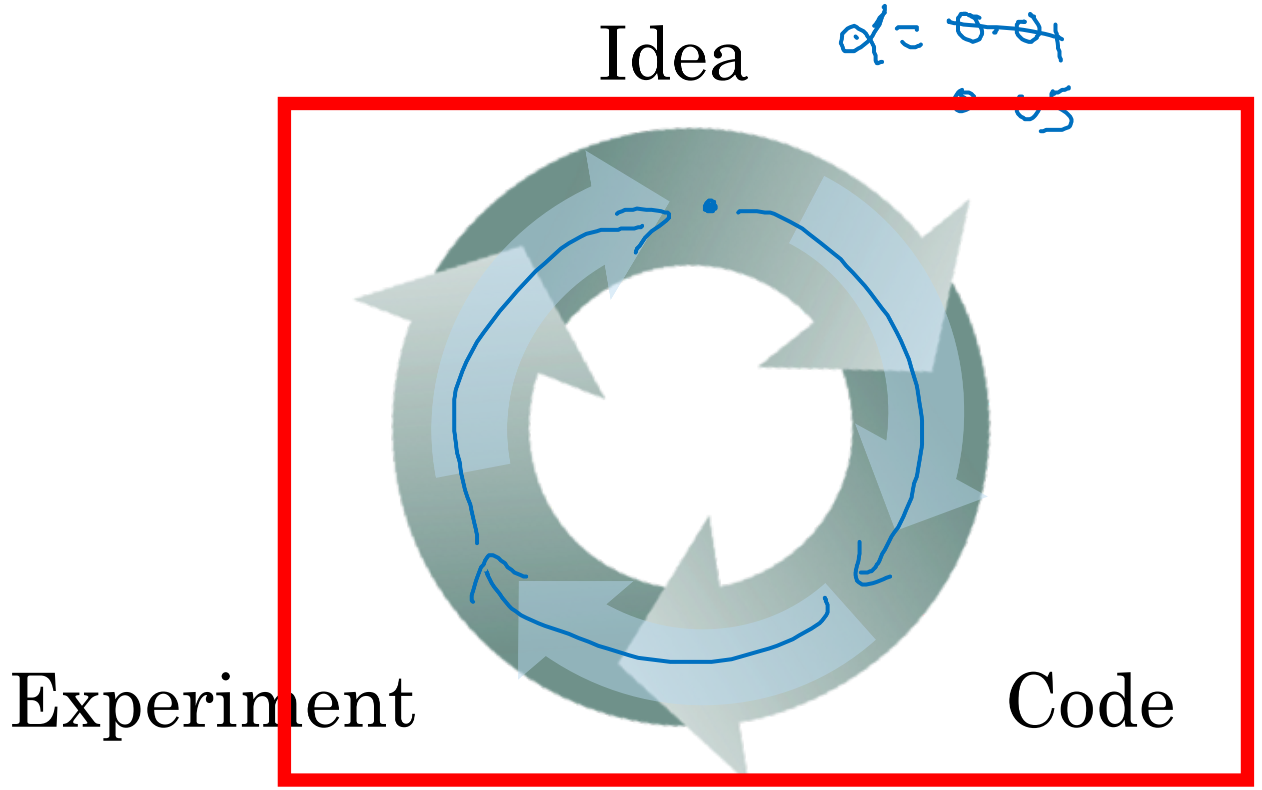
#hidden layers L

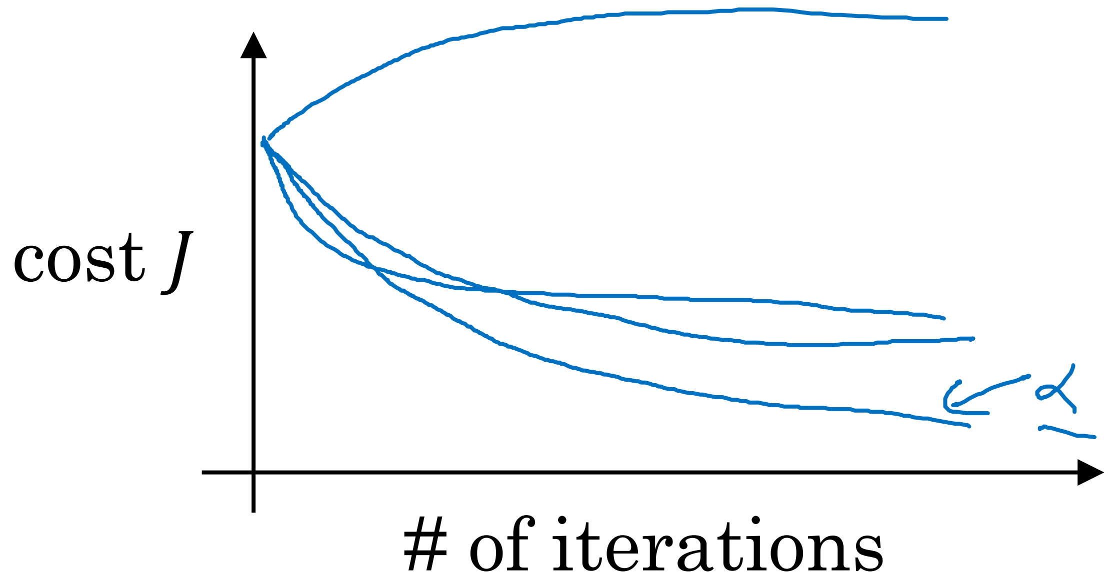
hidden units $n^{[1]}, n^{[2]}, \ldots$

choice of activation function

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

Applied deep learning is a very empirical process





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

deeplearning.ai

Deep Neural Networks

What does this have to do with the brain?

Forward and backward propagation

Note there are some typos here in

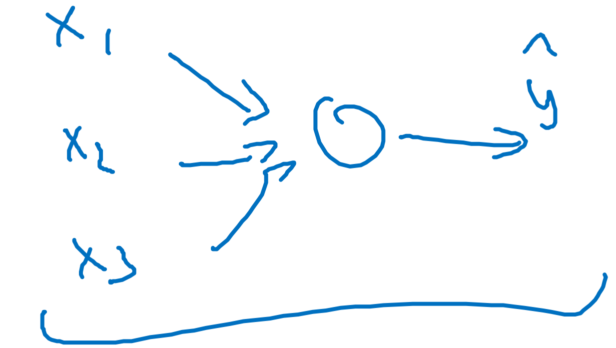
$$Z^{[1]} = W^{[1]}X + b^{[1]}$$
$$A^{[1]} = g^{[1]}(Z^{[1]})$$
$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$
$$A^{[2]} = g^{[2]}(Z^{[2]})$$
$$\vdots$$
$$A^{[L]} = g^{[L]}(Z^{[L]}) = \hat{Y}$$

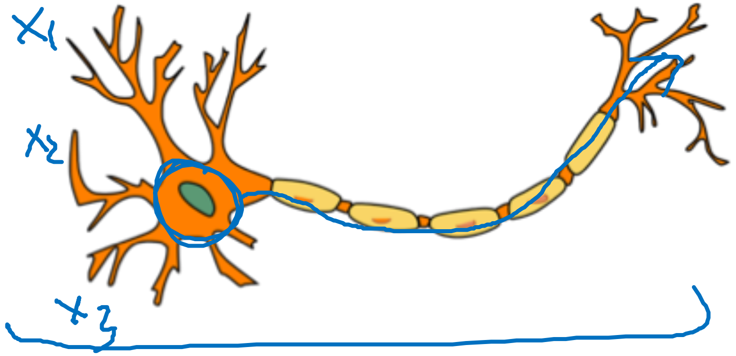
$$dZ^{[L]} = A^{[L]} - Y$$
$$dW^{[L]} = \frac{1}{m} dZ^{[L-1]} A^{[L]T}$$
$$db^{[L]} = \frac{1}{m} np.\mathrm{sum}(\mathrm{d}Z^{[L]}, axis = 1, keepdims = True)$$
$$dZ^{[L-1]} = dW^{[L]^T} dZ^{[L]} * g'^{[L]}(Z^{[L-1]})$$

dZ[l] or g' ke beech mein dot nahi hai e

$$\vdots$$
$$dZ^{[1]} = dW^{[2]^T} dZ^{[2]} g'^{[1]}(Z^{[1]})$$
$$dW^{[1]} = \frac{1}{m} dZ^{[1]} A^{[0]^T}$$
$$db^{[1]} = \frac{1}{m} np.\mathrm{sum}(\mathrm{d}Z^{[1]}, axis = 1, keepdims = True)$$

"It's like the brain"





$X \longrightarrow y$

Andrew Ng