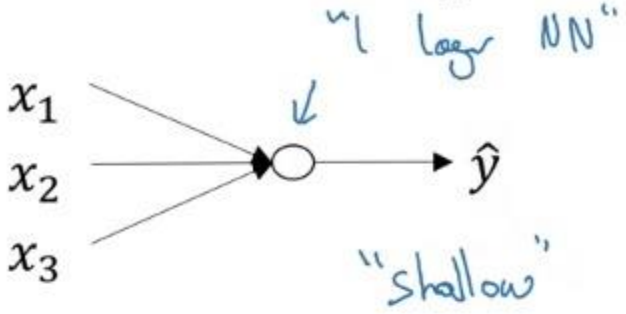
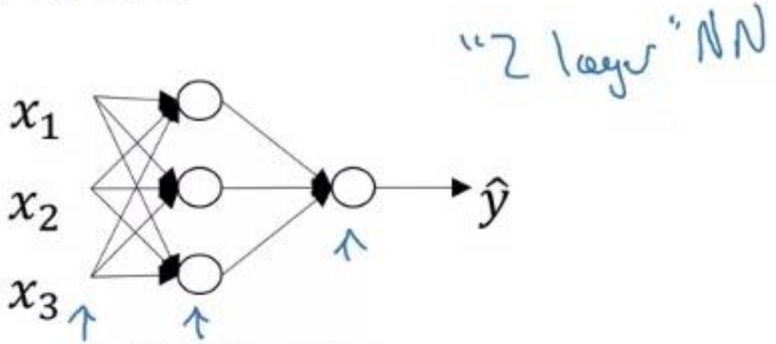
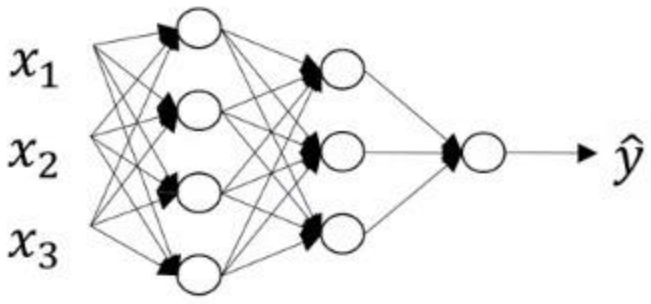
What is a deep neural network?



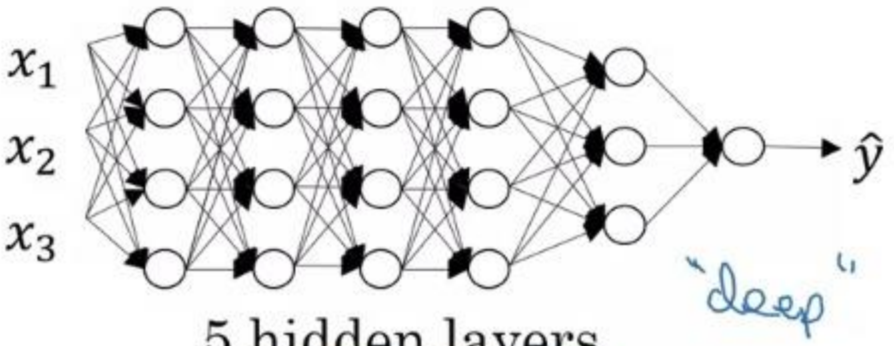
logistic regression



1 hidden layer



2 hidden layers

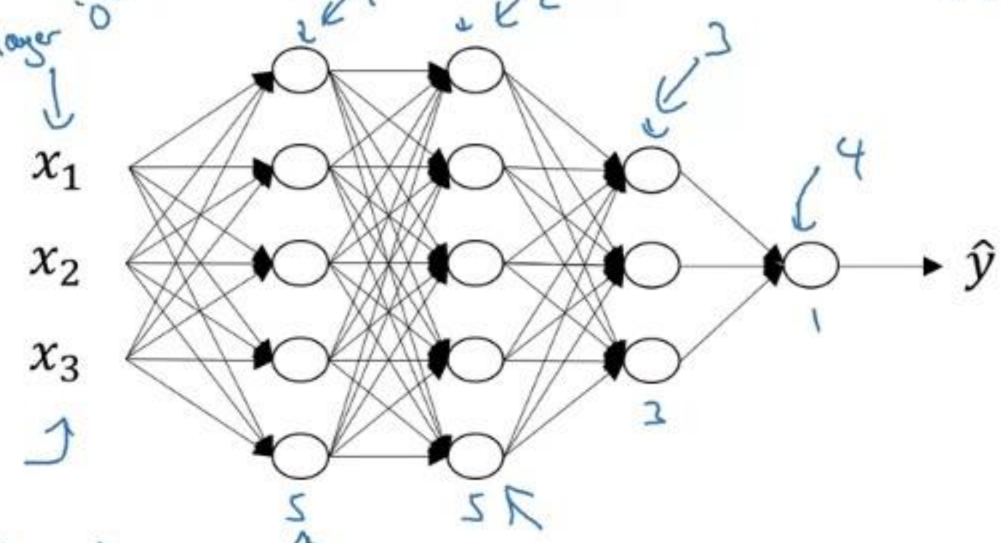


5 hidden layers

Andrew Ng

Deep neural network notation

4 layer NN



$X = a^{[0]}$

$\hat{y} = a^{[L]}$

$L = 4$ (#layers)

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

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

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

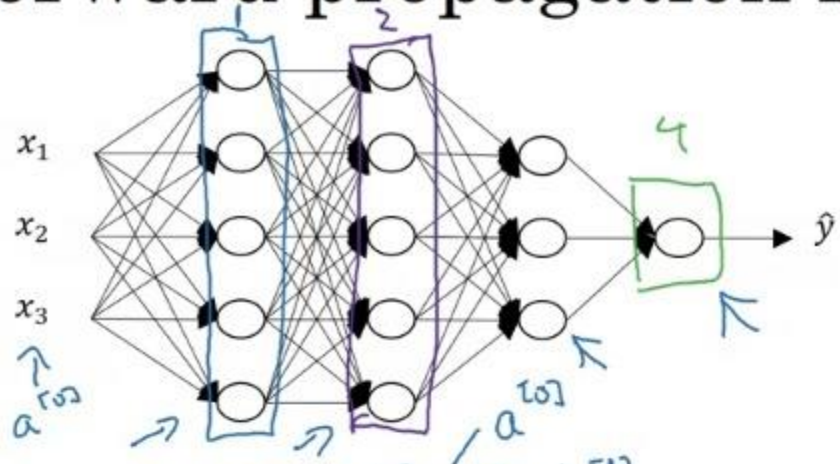
$a^{[l]}$ = activations in layer l

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

$b^{[l]}$

Andrew Ng

Forward propagation in a deep network



$X: \quad z^{[1]} = w^{[1]} a^{[0]} + b^{[1]}$

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

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

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

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

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

$A^{[0]} = X$

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

Vectorized:

$X = A^{[0]}$

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

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

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

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

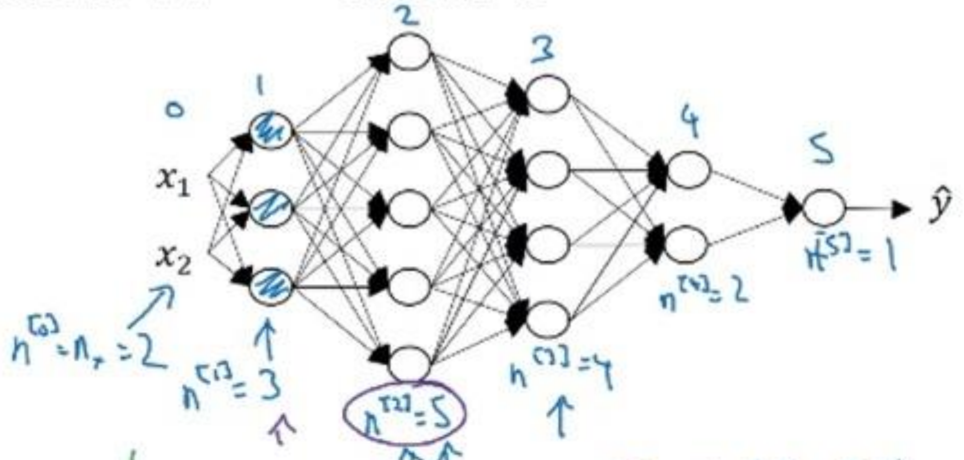
$\hat{Y} = g(Z^{[4]}) = A^{[4]}$

for $l = 1..4$

Andrew Ng

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

$L = 5$



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

$a^{[l]}$

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

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

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

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

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

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

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

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

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

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

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

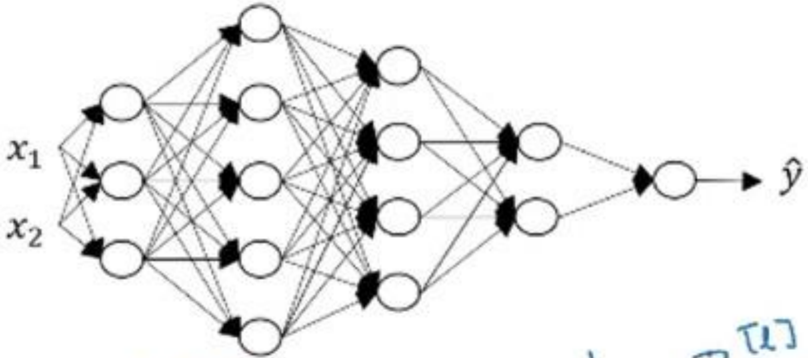
$\rightarrow (5,1) \quad (5,3) \quad (3,1) \quad (5,1)$

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

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

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

Vectorized implementation



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

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

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

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

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

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

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

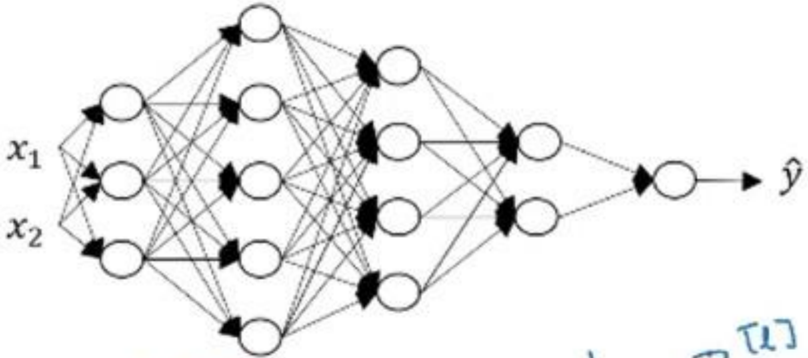
$Z^{[l]}, A^{[l]} : (n^{[l]}, m)$

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

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

Andrew Ng

Vectorized implementation



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

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

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

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

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

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

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

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

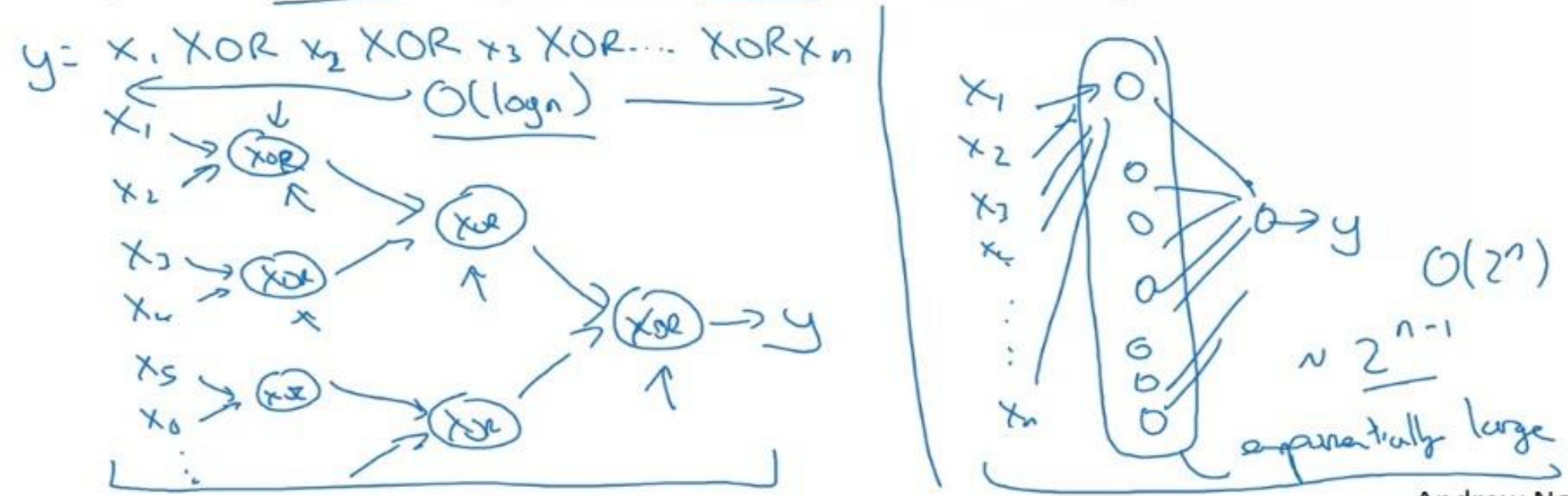
$l=0 \quad A^{[0]} = X = (n^{[0]}, m)$

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

Andrew Ng

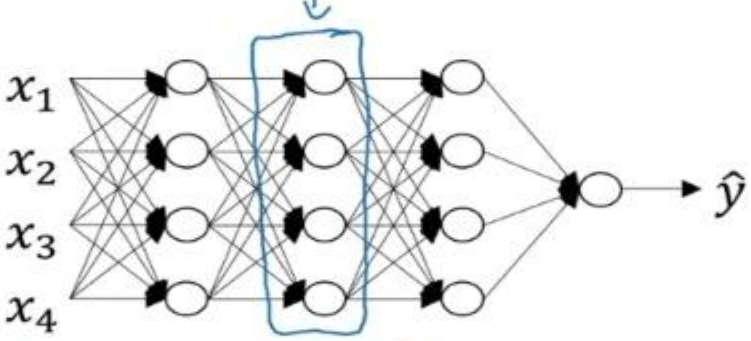
Circuit theory and deep learning

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

Forward and backward functions



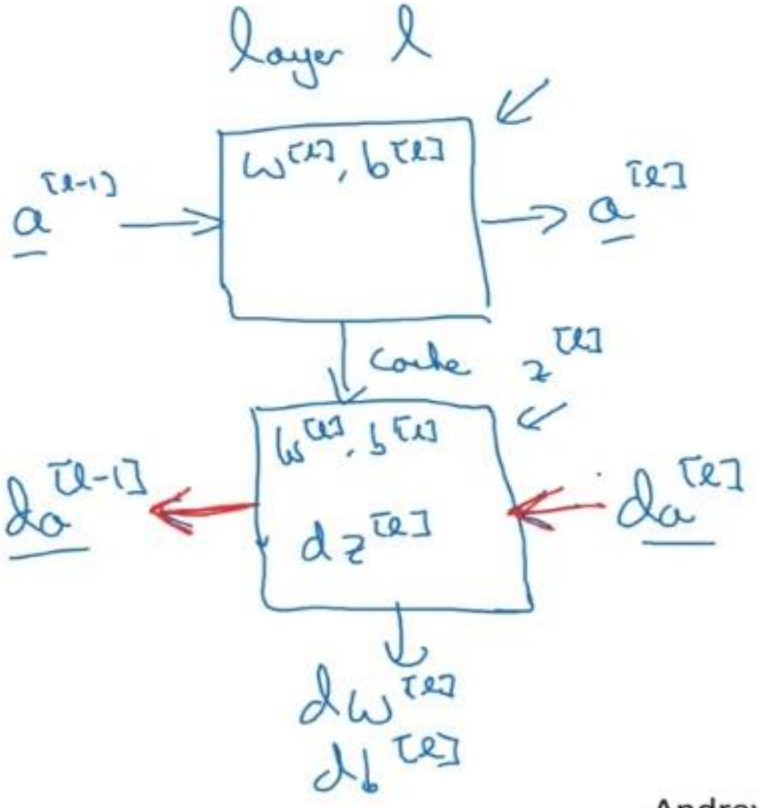
Layer l: $W^{[l]}, b^{[l]}$

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

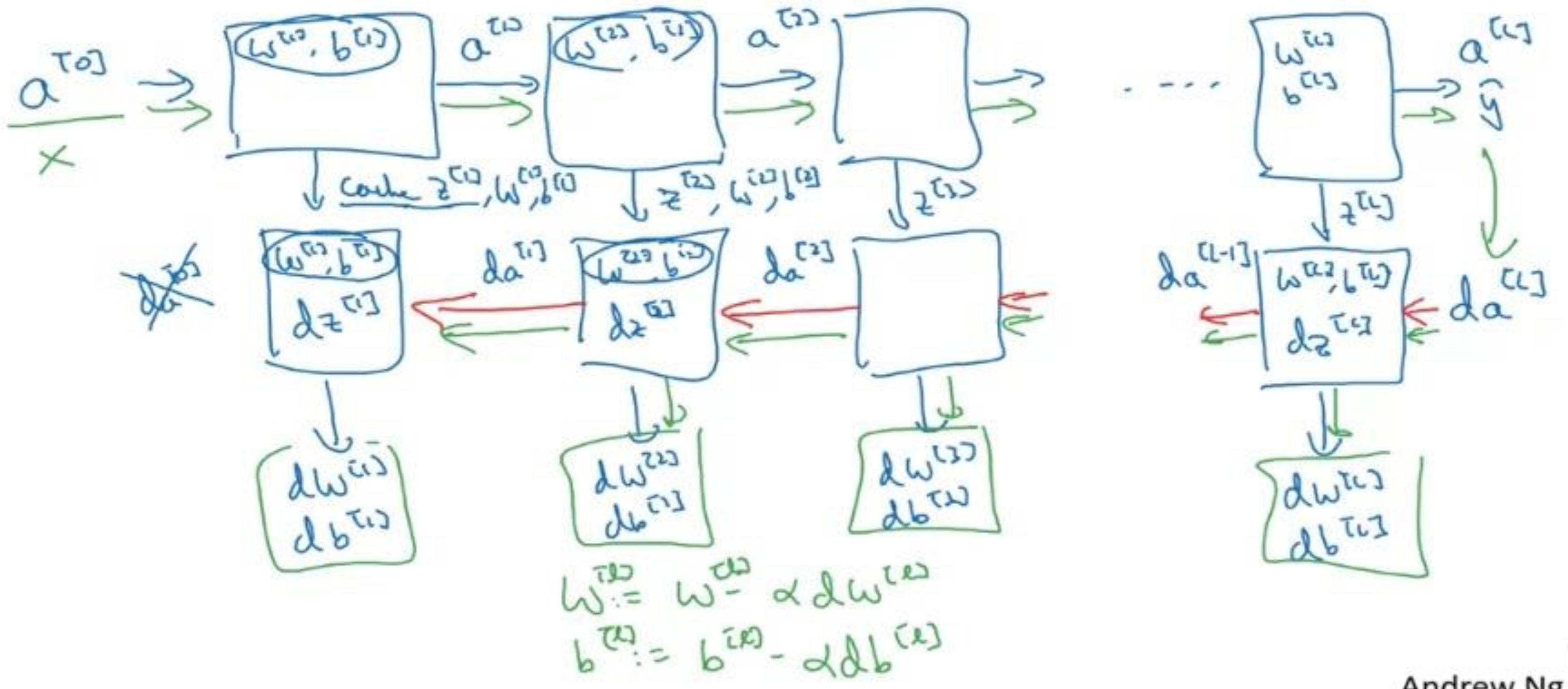
$\underline{z^{[l]}} = W^{[l]} \underline{a^{[l-1]}} + b^{[l]}$ cache $\underline{z^{[l]}}$

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

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



Forward and backward functions



Backward propagation for layer l

- Input $da^{[l]}$
- Output $\boxed{da^{[l-1]}}$, $dW^{[l]}$, $db^{[l]}$

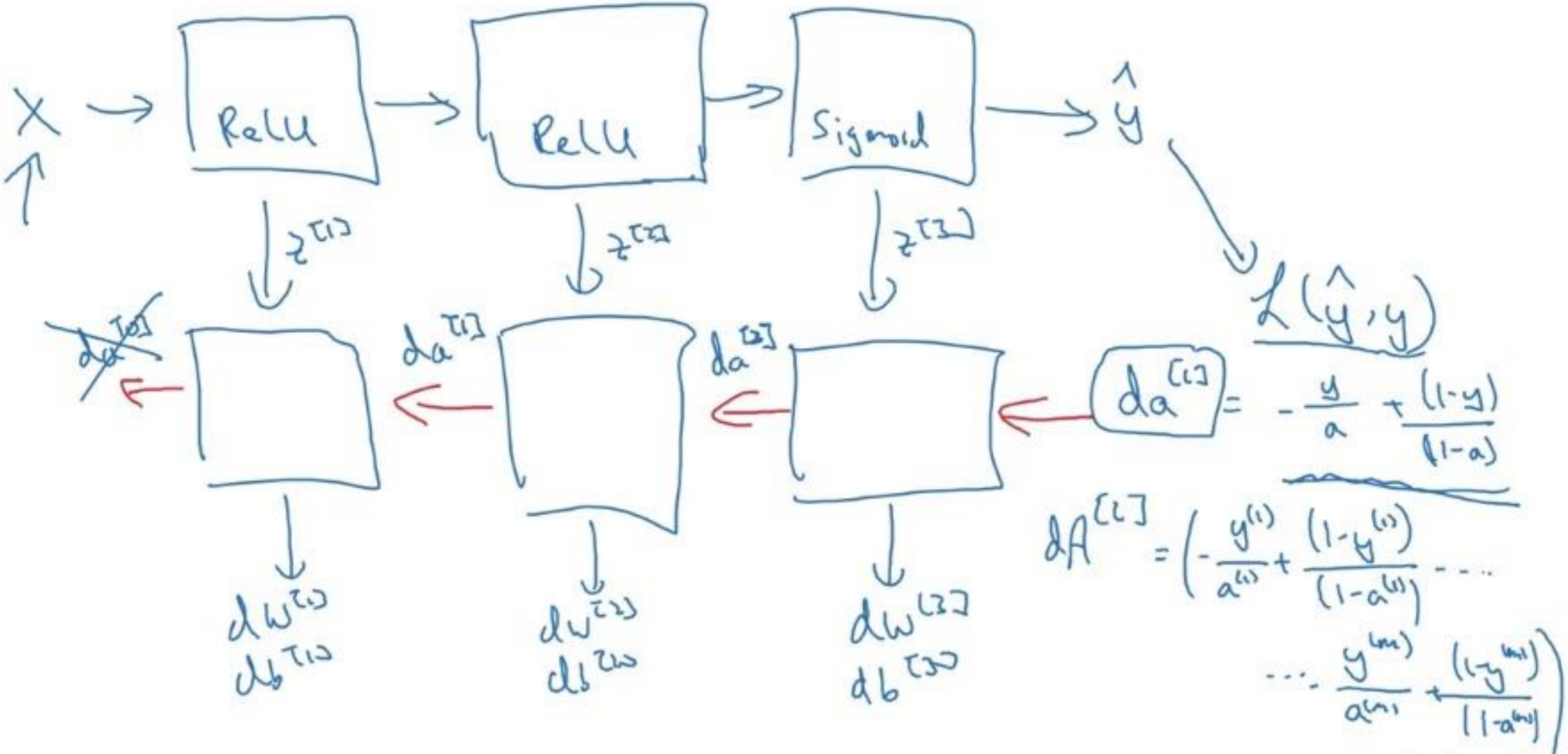
$$dz^{[l]} = da^{[l]} * g^{[l]\prime}(z^{[l]})$$
$$dW^{[l]} = dz^{[l]} \cdot a^{[l-1]}$$
$$db^{[l]} = dz^{[l]}$$
$$\boxed{da^{[l-1]}} = W^{[l]T} \cdot dz^{[l]}$$

$$dz^{[l]} = W^{[l+1]T} dz^{[l+1]} * g^{[l]\prime}(z^{[l]})$$

$$dZ^{[l]} = dA^{[l]} * g^{[l]\prime}(Z^{[l]})$$
$$dW^{[l]} = \frac{1}{m} dZ^{[l]} \cdot A^{[l-1]T}$$
$$db^{[l]} = \frac{1}{m} np.sum(dZ^{[l]}, axis=1, keepdims=True)$$
$$\boxed{dA^{[l-1]}} = W^{[l]T} \cdot dZ^{[l]}$$

Andrew Ng

Summary



Andrew Ng

What are hyperparameters?

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

Hyperparameters: learning rate α

#iterations

#hidden layers L

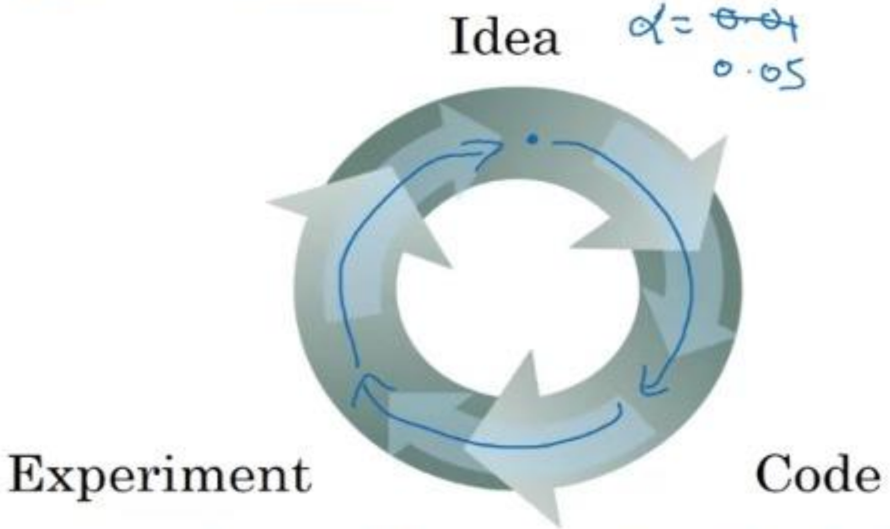
#hidden units $n^{[1]}, n^{[2]}, \ldots$

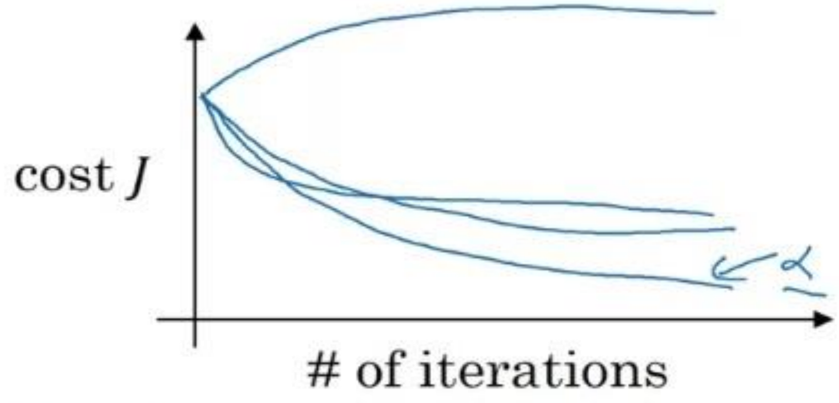
choice of activation function

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

Andrew Ng

Applied deep learning is a very empirical process





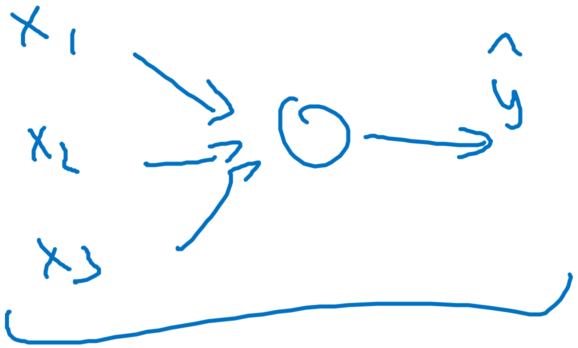
Vision, Speech, NLP, Ad, Search, Recommendation.

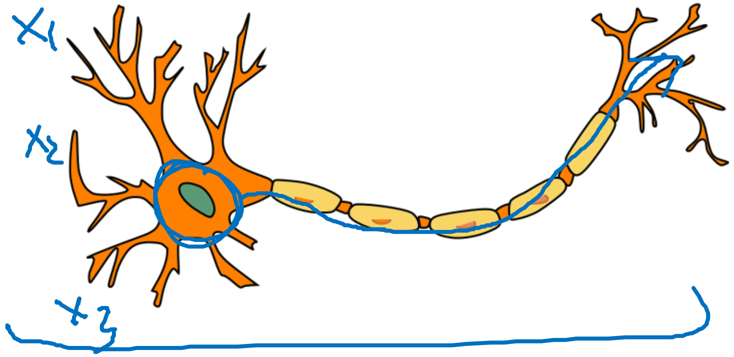
Forward and backward propagation

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

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

"It's like the brain."





$X \longrightarrow y$

Andrew Ng