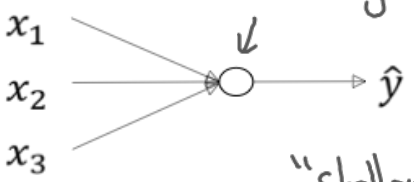
deeplearning.ai

Deep Neural Networks

Deep L-layer Neural network

What is a deep neural network?

"1 layer NN"

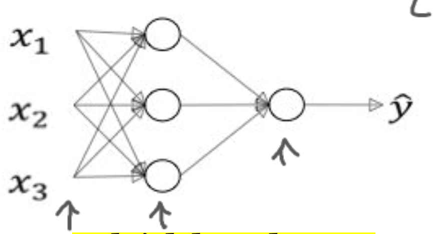


"shallow"

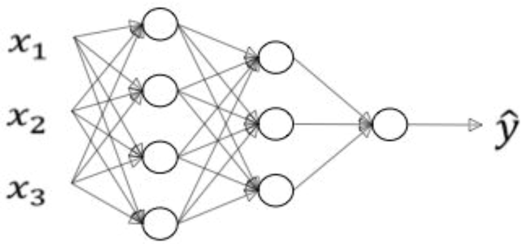
shallow Network

logistic regression

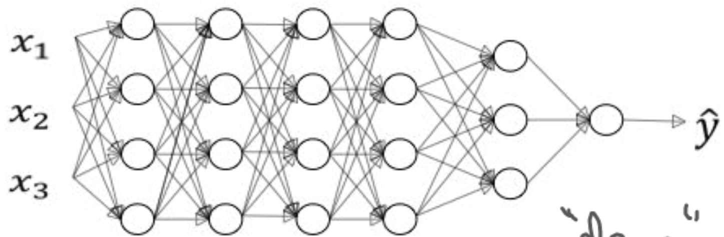
"2 layer" NN



1 hidden layer



2 hidden layers



"deep"

5 hidden layers

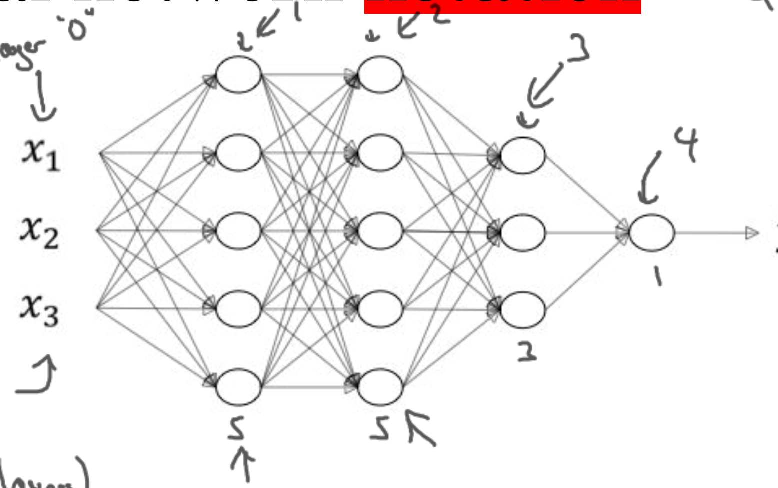
Deep Network

$$L,\ n^{[l]},\ a^{[l]},\ z^{[l]},\ W^{[l]},\ b^{[l]} \Rightarrow \begin{cases} X = a^{[0]} \\ \hat{y} = a^{[l]} \end{cases}$$

Deep neural network notation

4 layer NN

$X = a^{[0]}$



$L \rightarrow L = 4$ (#layers)

$n^{[l]} \rightarrow n^{[l]} =$ #units in layer l $\quad n^{[1]} = 5,\ n^{[2]} = 5,\ n^{[3]} = 3,\ n^{[4]} = n^{[L]} = 1$

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

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

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

$b^{[l]}$

$z^{[l]}$

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

deeplearning.ai

Deep Neural Networks

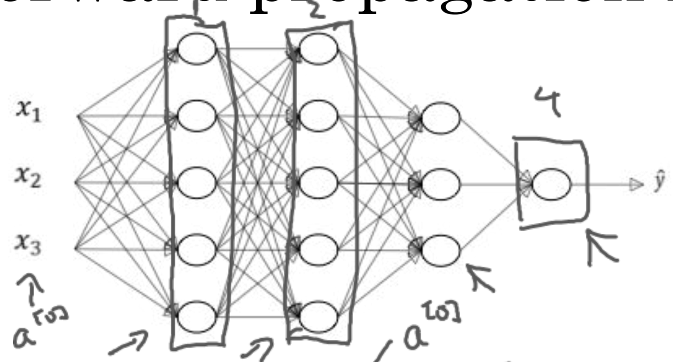
Forward Propagation in a Deep Network

$$z^{[l]} = W^{[l]} A^{[l]} + b^{[l]} \qquad A^{[l]} = g^{[l]}(z^{[l]})$$

$$\Rightarrow \begin{cases} X = A^{[0]} \\ \hat{Y} = A^{[l]} \end{cases}$$

for $l=1$ to $l=L$

Forward propagation in a deep network



$A^{[0]} = X$

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

x: $z^{[1]} = w^{[1]} a^{[0]} + b^{[1]}$

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

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

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

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

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

Vectorized:

$X = A^{[0]}$

for $l = 1..4$

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

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

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

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

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

deeplearning.ai

Deep Neural Networks

Getting your matrix dimensions right

Dimensions

non_vectorized (one sample)

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

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

Vectorized version (m samples)

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

→ $n^{[l]}$ = number of units in layer l

→ $\dim(x) = \dim(dx) \Rightarrow dw^{[l]} = (n^{[l]}, n^{[l-1]}), \; db^{[l]} = (n^{[l]}, 1)$

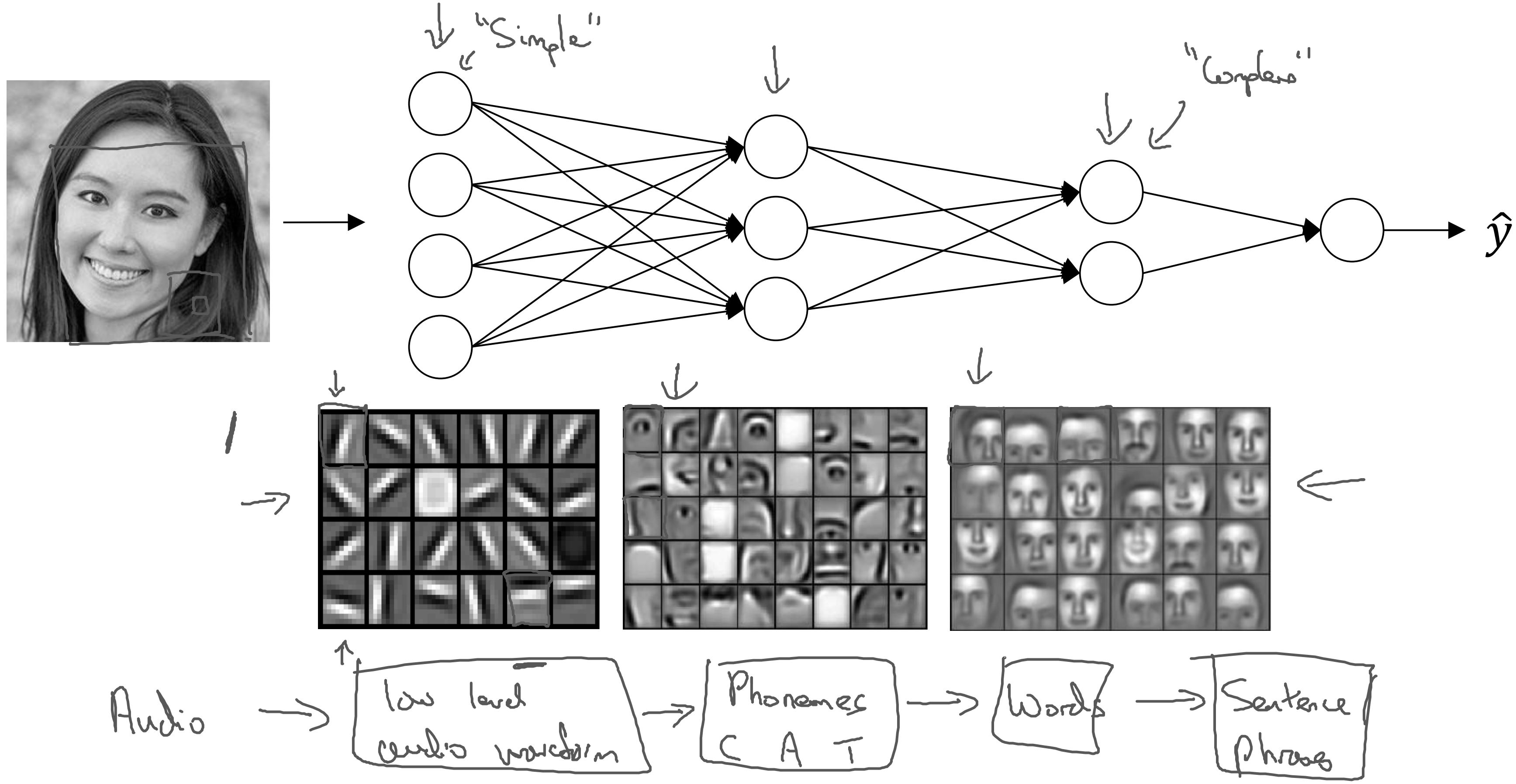
→ $X = A^{[0]} \Rightarrow X = (n^{[0]}, m)$

deeplearning.ai

Deep Neural Networks

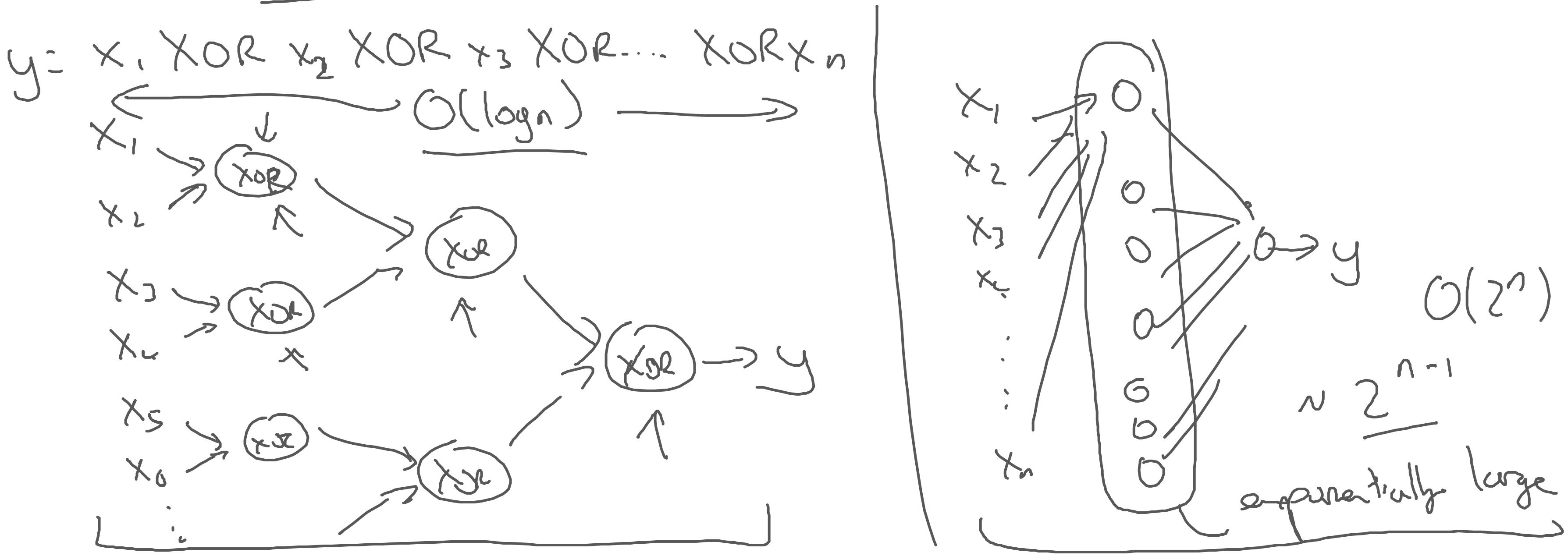
Why deep representations?

Intuition about deep representation



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.

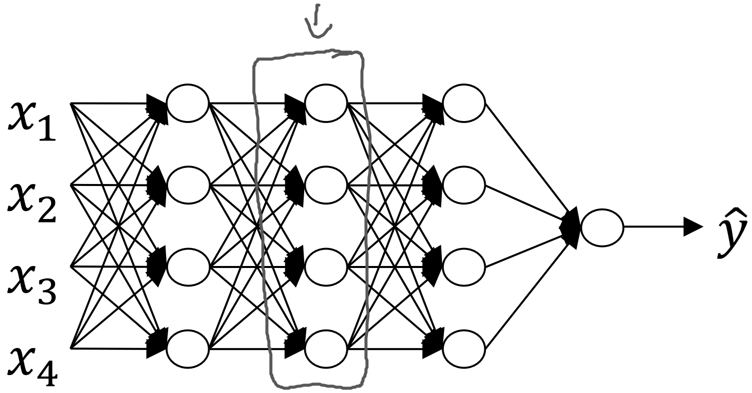


deeplearning.ai

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 $a^{[l]}$

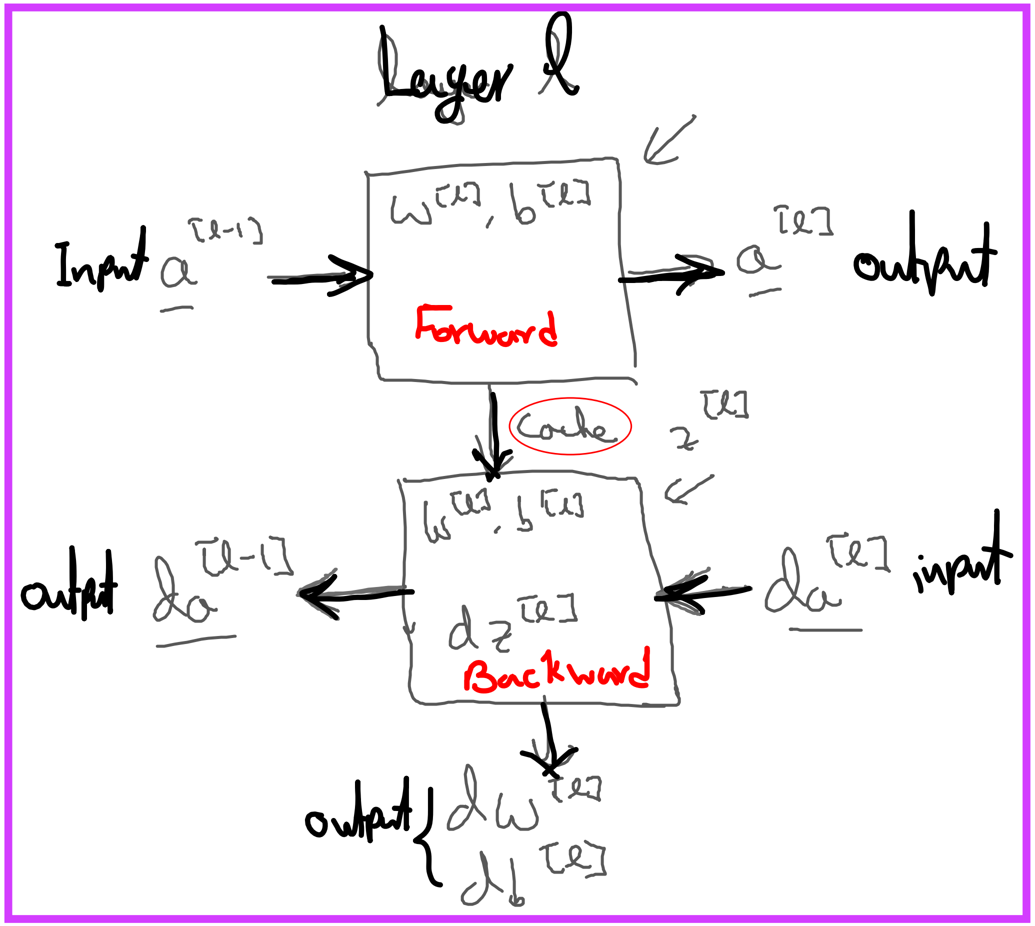
$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$ cache $z^{[l]}$

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

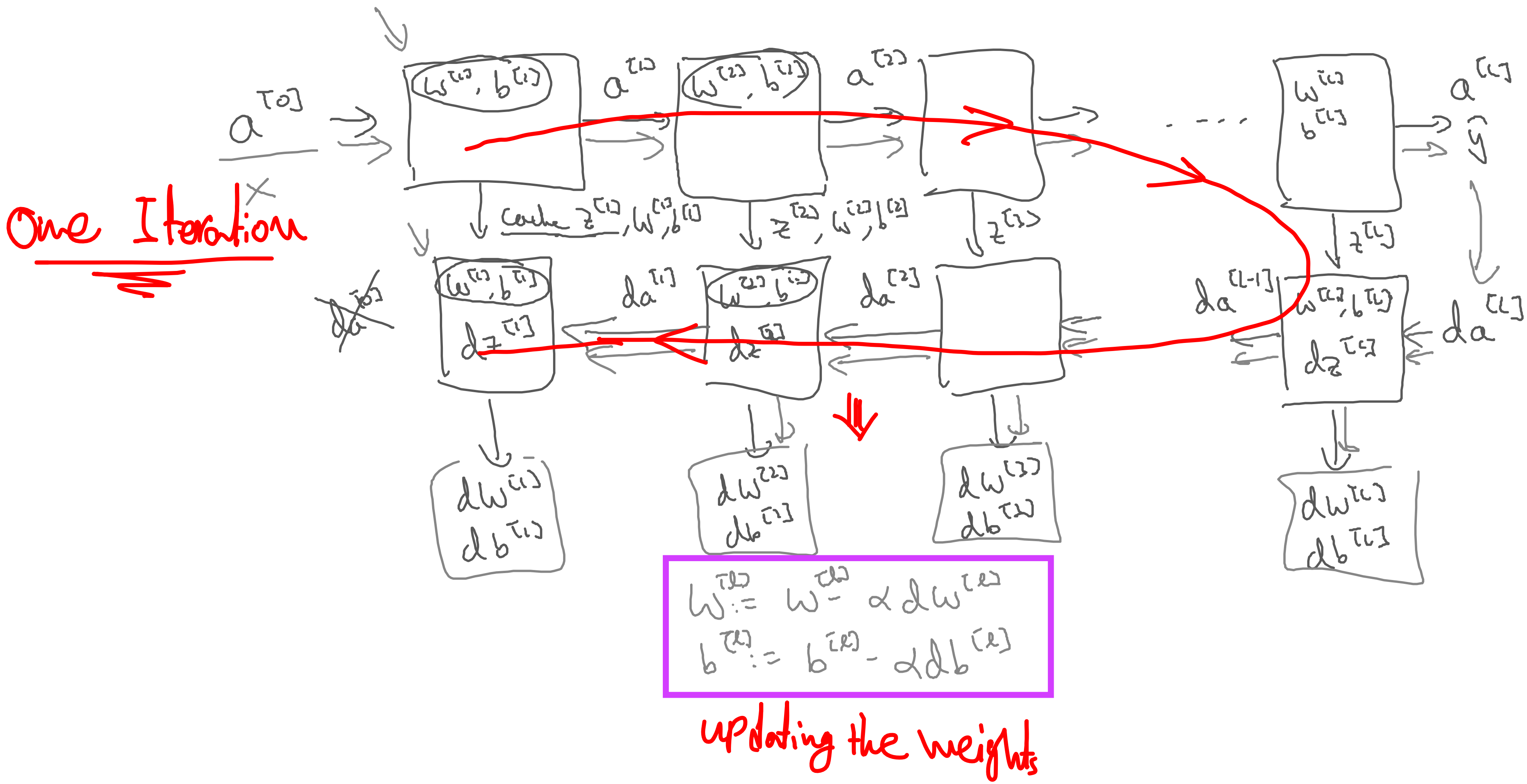
→ Backward: Input $da^{[l]}$, cache($z^{[l]}$) output $da^{[l-1]}$

$dW^{[l]}$

$db^{[l]}$



Forward and backward functions



deeplearning.ai

Deep Neural Networks

Forward and backward propagation

Forward propagation for layer l

Input $a^{[l-1]}$

Output $a^{[l]}$, cache ($z^{[l]}$)

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

Vectorized Version

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

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

Vectorized:

$$Z^{[l]} = W^{[l]} \cdot A^{[l-1]} + b^{[l]}$$

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

$a^{[0]}$

$A^{[0]}$

$$X = A^{[0]} \rightarrow \square \rightarrow \square \rightarrow \square \rightarrow$$

Backward propagation for layer l

→ Input $da^{[l]}$

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

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

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

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

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

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

Vectorized version

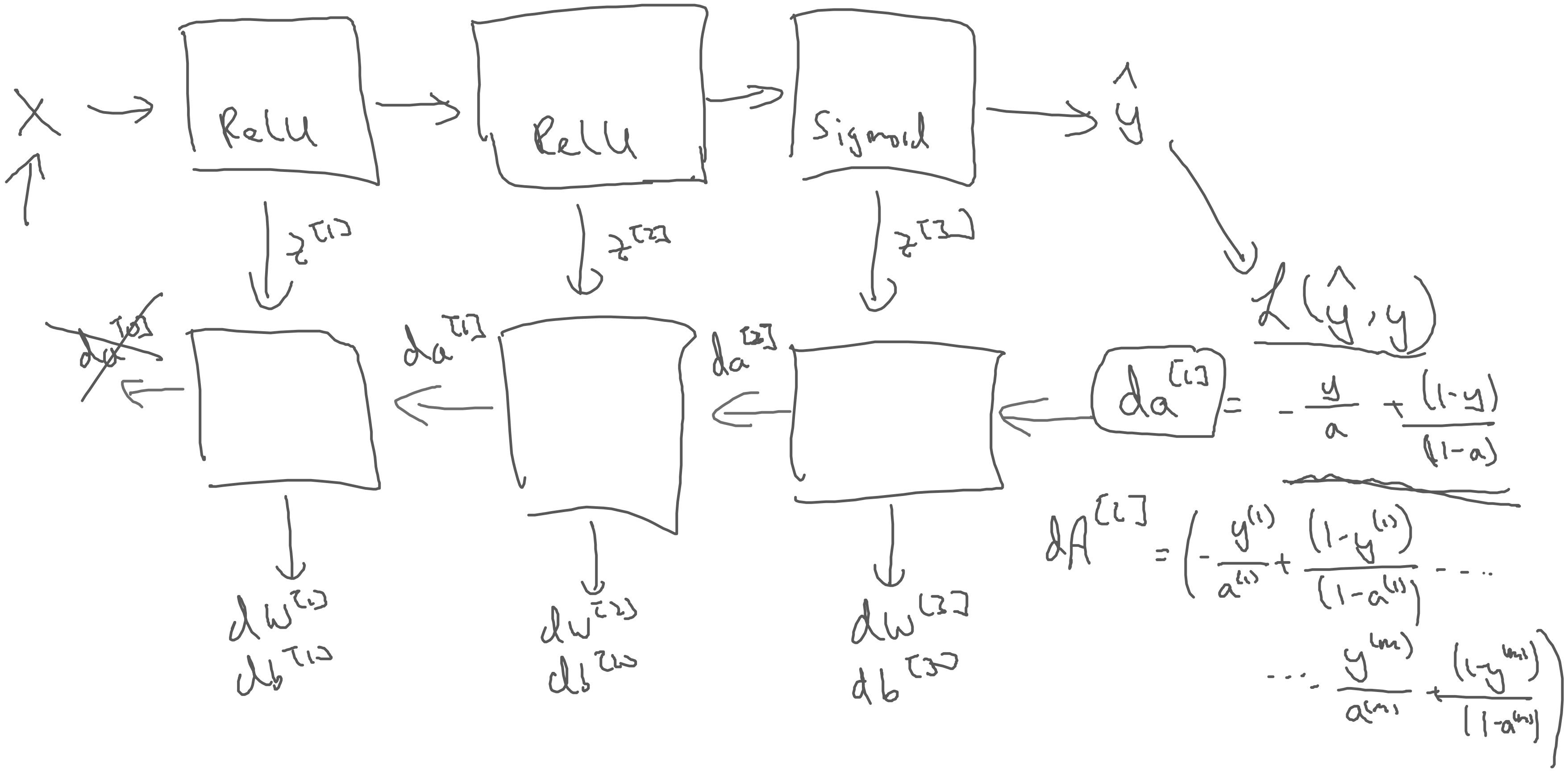
$$dZ^{[l]} = dA^{[l]} * g^{[l]\prime}(Z^{[l]})$$

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

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

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

Summary



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

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

Examples of Hyperparameters

→ #iterations

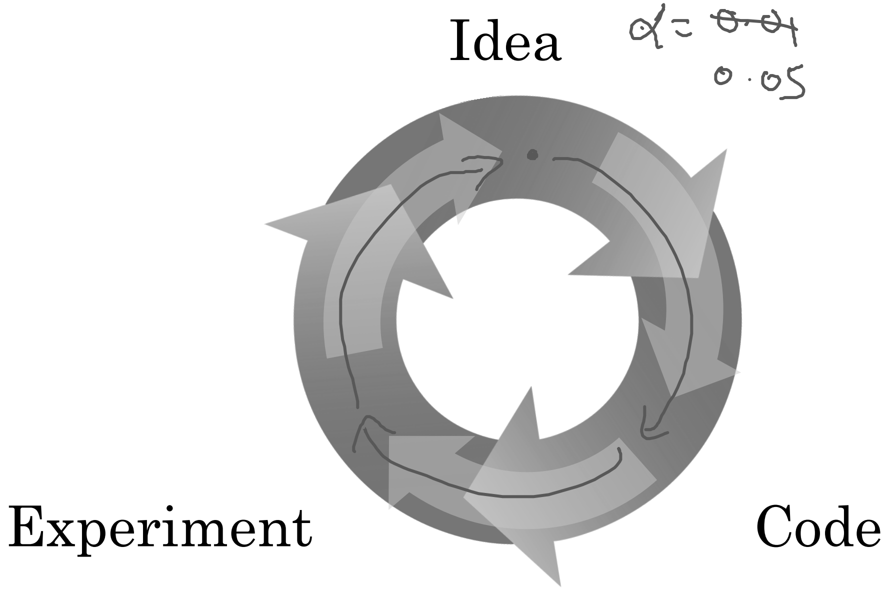
→ #hidden layers L

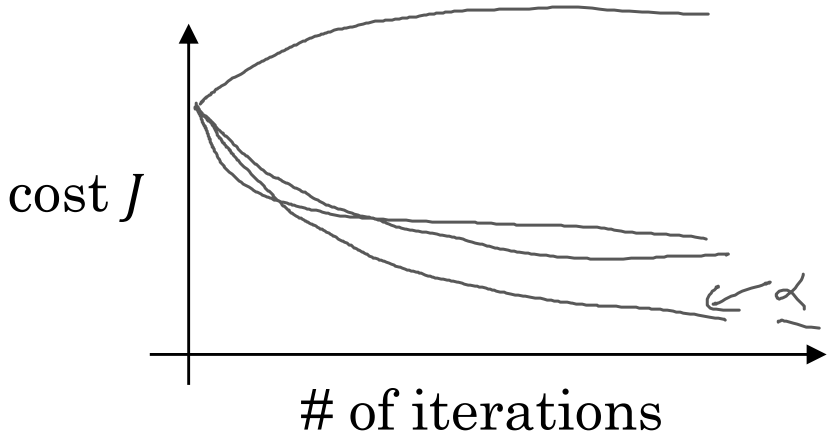
→ # hidden units $n^{[1]}, n^{[2]}, \ldots$

→ 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

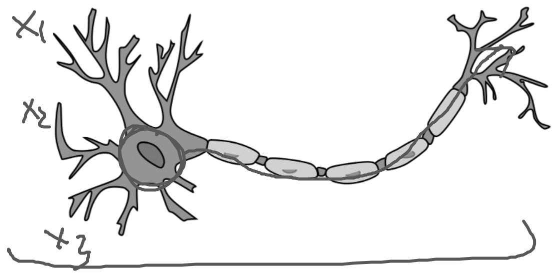
What does this have to do with the brain?

Forward and backward propagation

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

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

"It's like the brain"



$X \longrightarrow y$