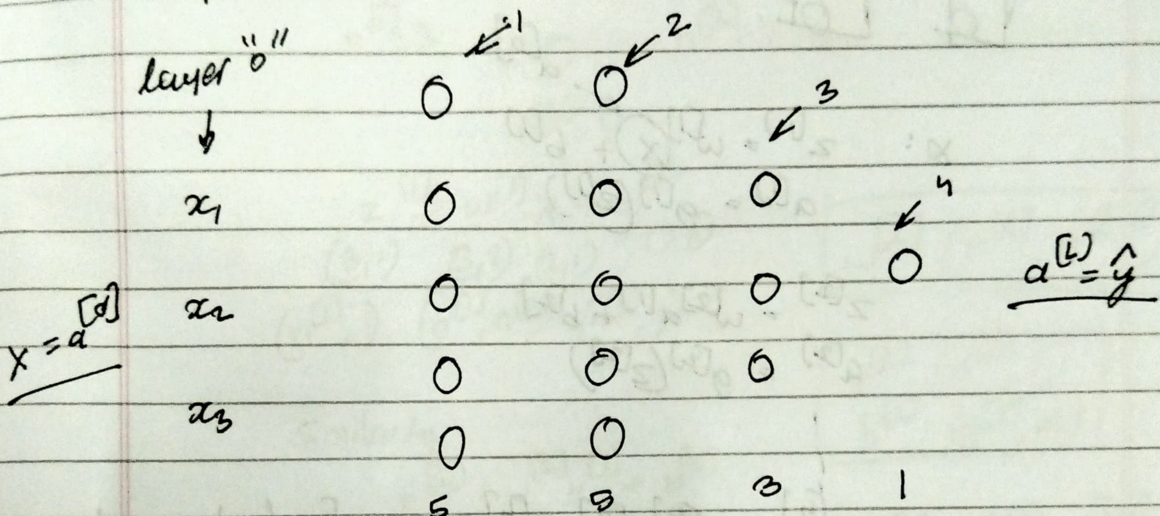
DEEP NEURAL NETWORK

what is deep neural network

logistic regression	Shallow
1 hidden layer	↓
2 hidden layer	↓
5 hidden layers	Deep

Deep neural network notation



→ $L = 4$ (# layers)

→ $n^{[l]}$ = # units in layer l $\quad n^{1}$

$n^{[0]}$ $\quad n^{[1]} = 5, \; n^{[2]} = 5, \; n^{[3]} = 3, \; n^{[4]} = n^{[L]} = 1$

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

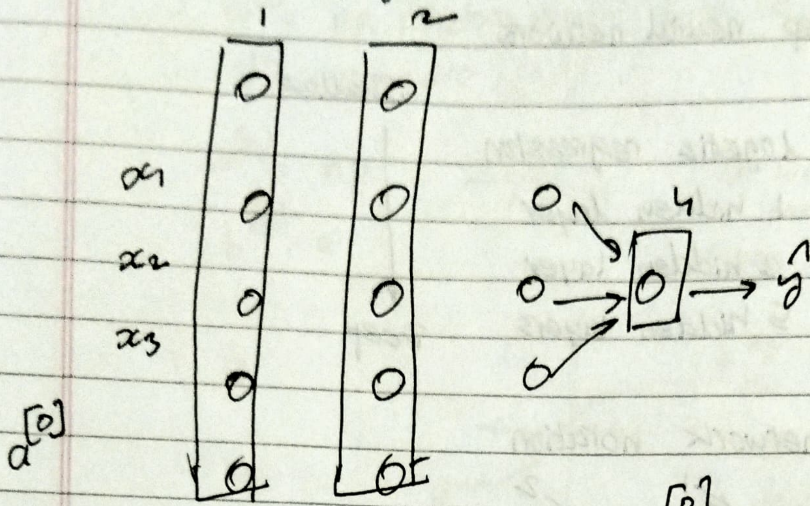
→ $a^{[l]}$ = activations in layer l

→ $a^{[l]} = g^{[l]}(z^{[l]})$

→ $w^{[l]}$ = weights for $z^{[l]}$

→ $b^{[l]}$

Forward propagation in Deep network



$$x: \quad z^{[1]} = w^{[1]} x + b^{[1]} \quad \leftarrow a^{[0]}$$
$$a^{[1]} = g^{[1]}(z^{[1]})$$

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

$$\vdots$$

$$z^{[4]} = w^{[4]} a^{[3]} + b^{[4]}$$
$$a^{[4]} = g^{[4]}(z^{[4]})$$

$$Z = \begin{bmatrix} | & | & | \\ z^{[2](1)} & z^{[2](2)} & z^{[2](3)} \\ | & | & | \end{bmatrix}$$

$$\boxed{\begin{aligned} z^{[l]} &= w^{[l]} a^{[l-1]} + b^{[l]} \\ a^{[l]} &= g^{[l]}(z^{[l]}) \end{aligned}}$$

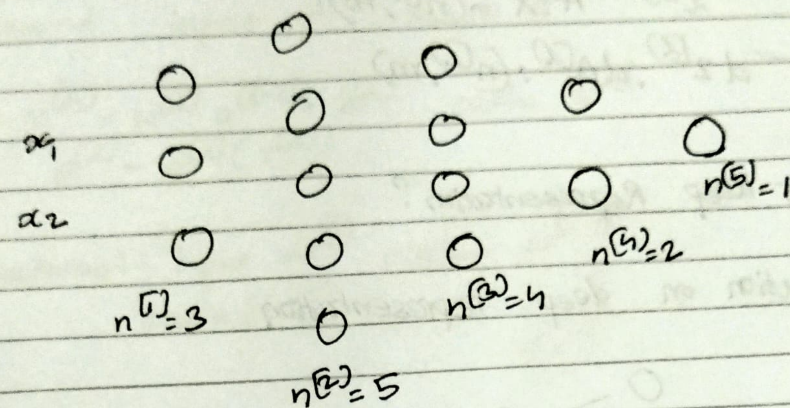
vertical

$$\rightarrow z^{[1]} = w^{[1]} A^{[0]} + b^{[1]}$$
$$A^{[1]} = g^{[1]}(z^{[1]})$$
$$\rightarrow z^{[2]} = w^{[2]} A^{[1]} + b^{[1]}$$
$$\rightarrow A^{[2]} = g^{[2]}(z^{[2]})$$
$$\hat{y} = g(z^{[4]}) = A^{[4]}$$

for $l = 1..4$

Getting your Matrix Dimensions Right

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



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

$(3,1)$ $(3,2)$ $(2,1)$ $(3,1)$

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

$\boxed{w^{[l]} : (n^{[l]}, n^{[l-1]})}$

$\boxed{z^{[l]} : (n^{[l]},1) \quad b^{[l]} : (n^{[l]},1)}$

Similarly

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

$(5,1)$ $(5,3)$ $(3,1)$ $(5,1)$

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

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

$w^{[3]} = (4,5)$

$w^{[4]} = (2,4)$

$w^{[5]} = (1,2)$

$\boxed{a^{[l]} = g^{[l]}(z^{[l]}) \quad a \,\&\, z \text{ have } (n^{[l]},1)}$

Vectorised Implementation

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

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

$$\begin{bmatrix} | & | & | \\ z & z & z \\ | & | & | \end{bmatrix} \rightarrow Z^{[1]} = W^{[1]} X + b^{[1]}$$

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

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

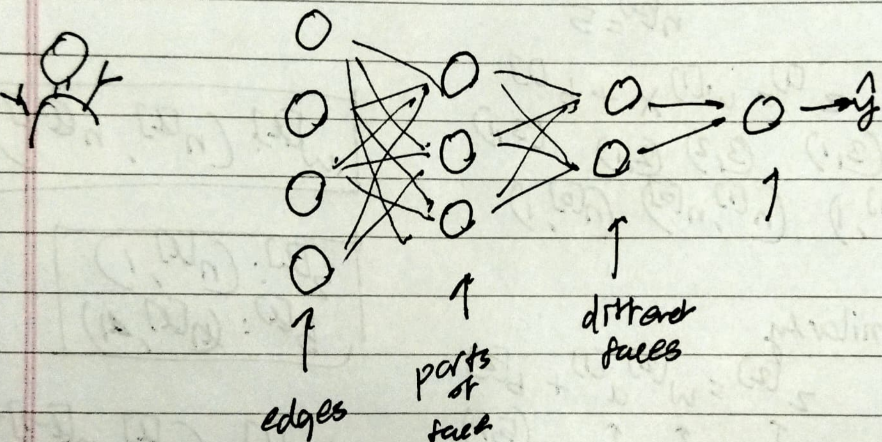
$Z^{[l]}, A^{[l]} : (n^{[l]}, m)$

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

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

Why deep Representation?

Intuition on deep representation



Circuit theory & deep learning.

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

Building blocks for ~~a~~ Deep learning.

layer l : $w^{[l]}, b^{[l]}$

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

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

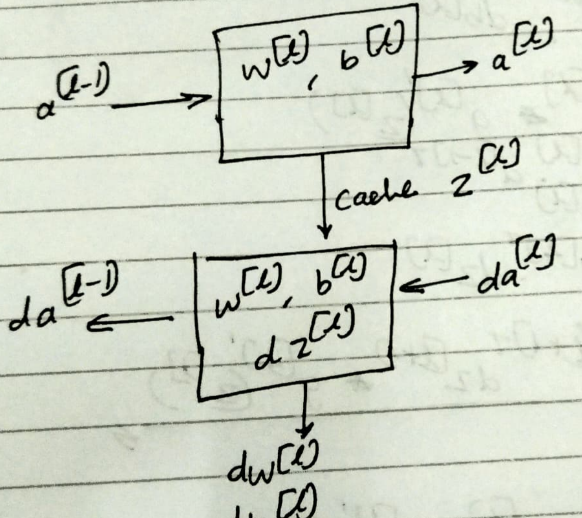
$a^{[l]} = g^{[l]}(z^{[l]})$

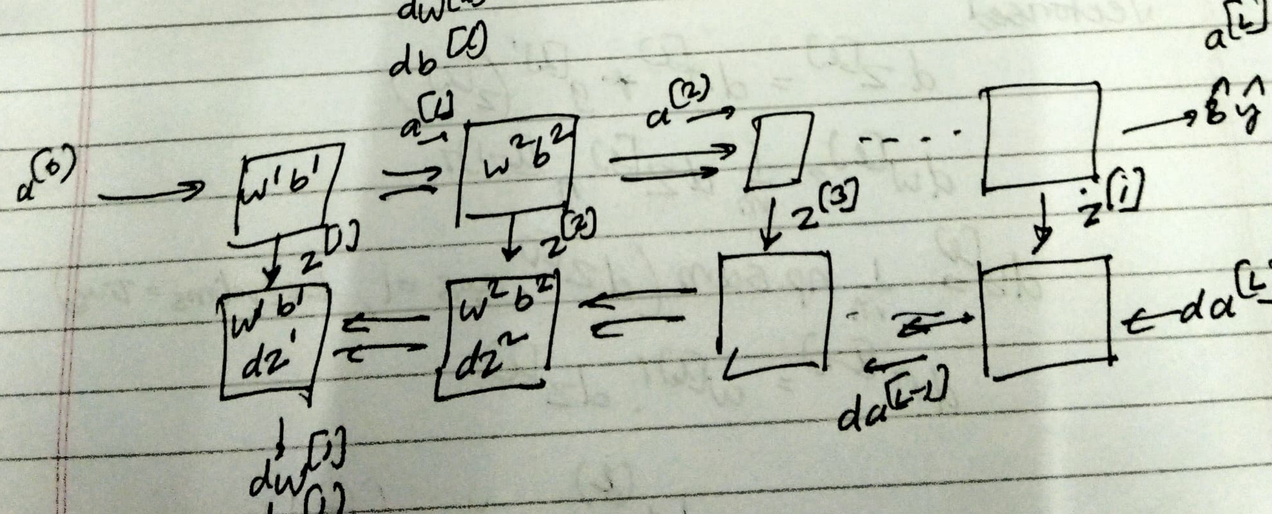
⇒ Backward : Input $da^{[l]}$, output $da^{[l-1]}$

cache $(z^{[l]})$ $dw^{[l]}$

$db^{[l]}$

layer l





⇒ Forward propagation for layer l

Input $a^{[l-1]}$
Output $a^{[l]}$ cache $z^{[l]}$

$$z^{[l]} = w^{[l]} a^{[l-1]} + b^{[l]}$$
$$a^{[l]} = g^{[l]}(z^{[l]})$$

Vectorized:

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$$
$$A^{[l]} = g^{[l]}(Z^{[l]})$$

⇒ Backword propagation for layer l

Input $da^{[l]}$
Output $da^{[l-1]}$, $dw^{[l]}$, $db^{[l]}$

$$dz^{[l]} = da^{[l]} * g^{[l]'}(z^{[l]})$$
$$dw^{[l]} = dz^{[l]} a^{[l-1]T}$$
$$db^{[l]} = dz^{[l]}$$
$$da^{[l-1]} = w^{[l]T} \cdot dz^{[l]}$$

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

Vectorised

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

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

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

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

$$da^{(l)} = -\frac{y}{a} + \left(\frac{1-y}{1-a}\right)$$

Parameters vs Hyperparameters

Parameters : $w^{[1]}, b^{[1]}, w^{[2]}, b^{[2]}, w^{[3]}, b^{[3]}, \ldots$

Hyper parameters
- learning rate α
- #iterations
- #hidden layers
- #hidden units $n^{[1]}, n^{[2]}, \ldots$
- choice of activation functions

Later : Momentum, minibatch, size, · regulations

Applied deep learning

⇓

Emperical process

(Trying till something clicks)