Deep Neural Network:

→ We've seen fwd, bckwd propogation for a single layer, log reg & vectorization. We saw why its important to initialize vectors randomly. We'll use these together for a deeper NN.

Deep NN:

Log reg: kinda shallow

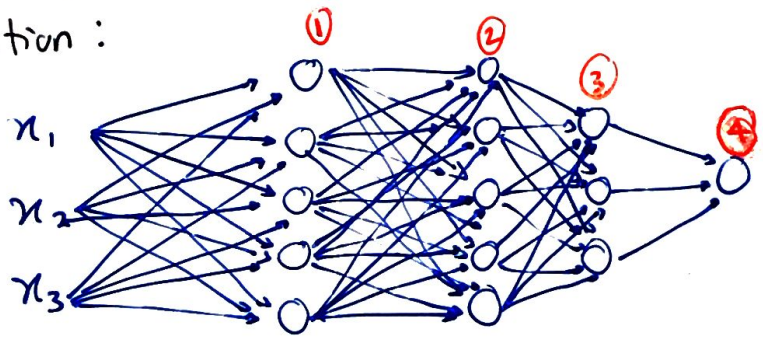
1 hidden layer: better but basic

Deeper NNs? functions that can be learned here, that shallower Networks are unable to recognize.

New Problems to think?

→ depth needed for a NN to work optimally : New hyperparameter

Notation:



→ 4 layed NN $L=4$

→ 3 hidden layers

→ 5, 5, 3

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

$n^{[1]} = 5$ $n^{[3]} = 3$

$n^{[2]} = 5$ $n^{[4]} = 1$

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

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

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

Forward propagation:

for layer 1,

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

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

$a^{[2]} = g^{[2]}(z^{[2]})$ → ... → $z^{[4]} = w^{[4]} a^{[3]} + b^{[4]}$

$a^{[4]} = g^{[4]}(z^{[4]}) = \hat{y}$

General form?

Going by the pattern $l: 1 \to 4$, we see,

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

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