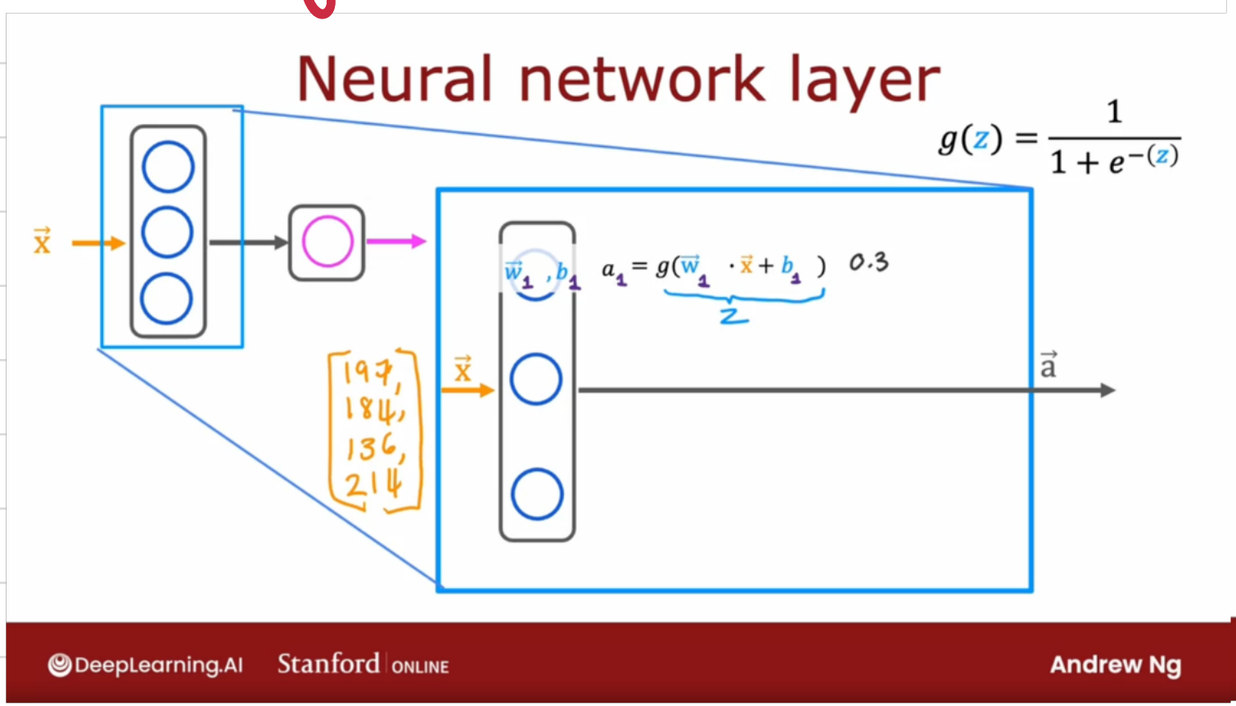
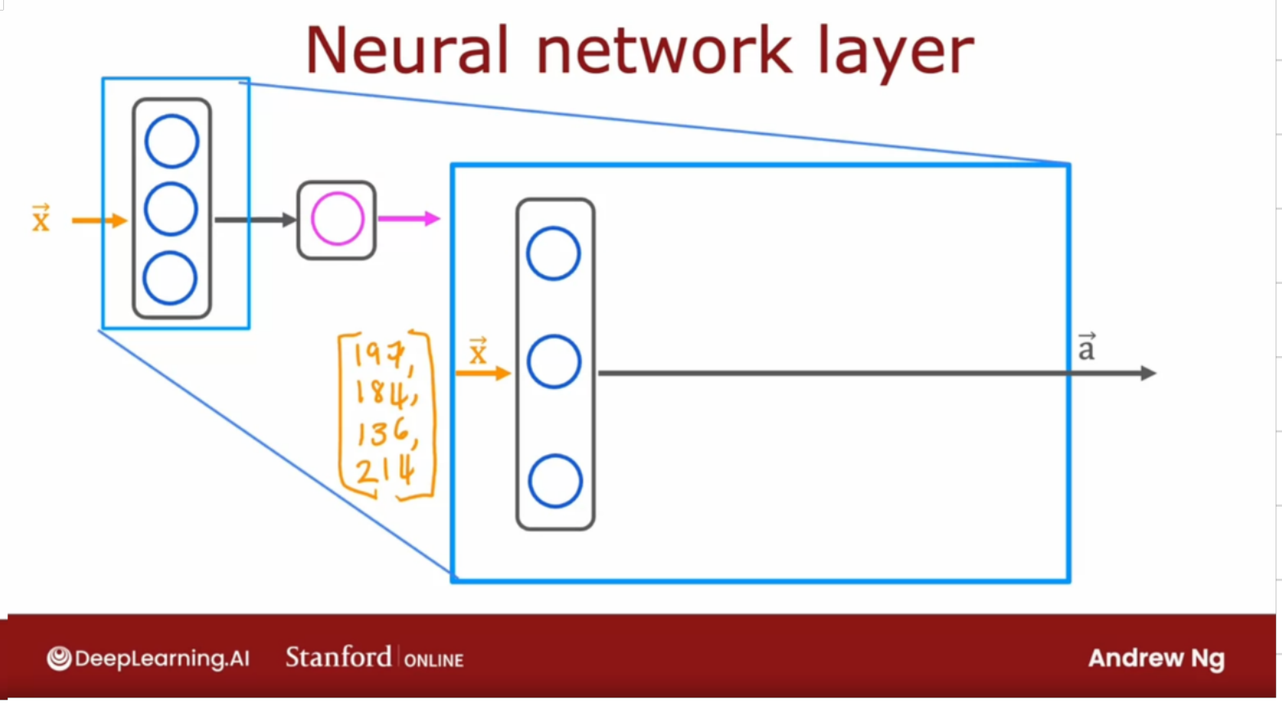
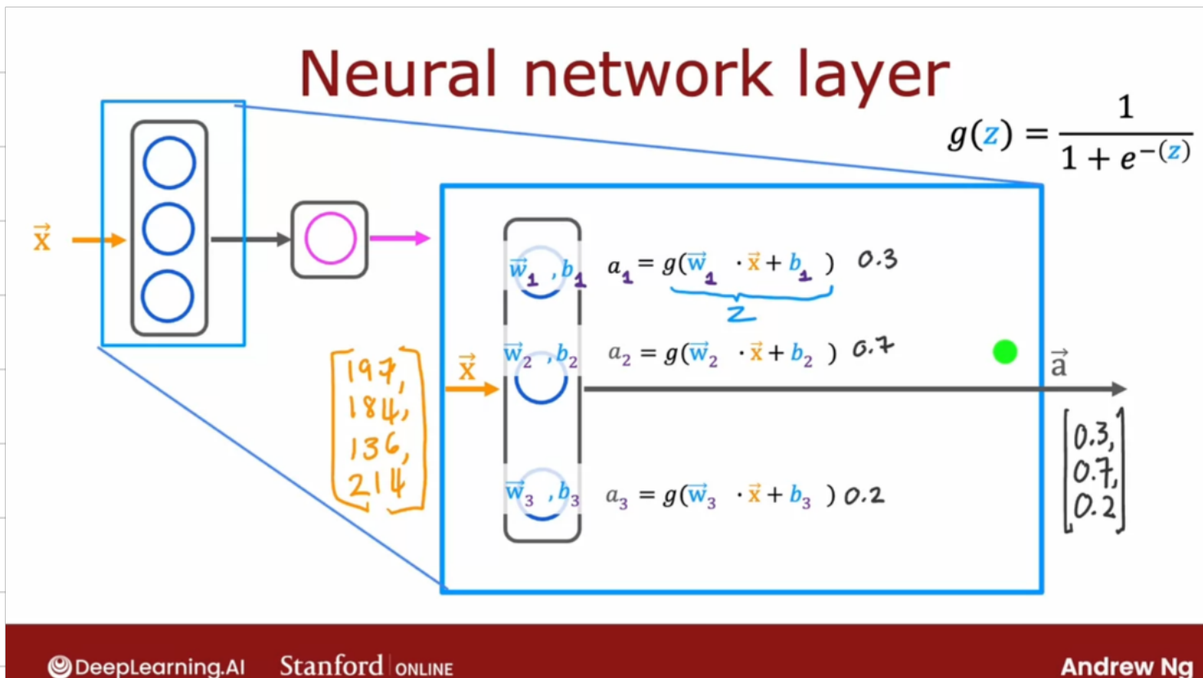
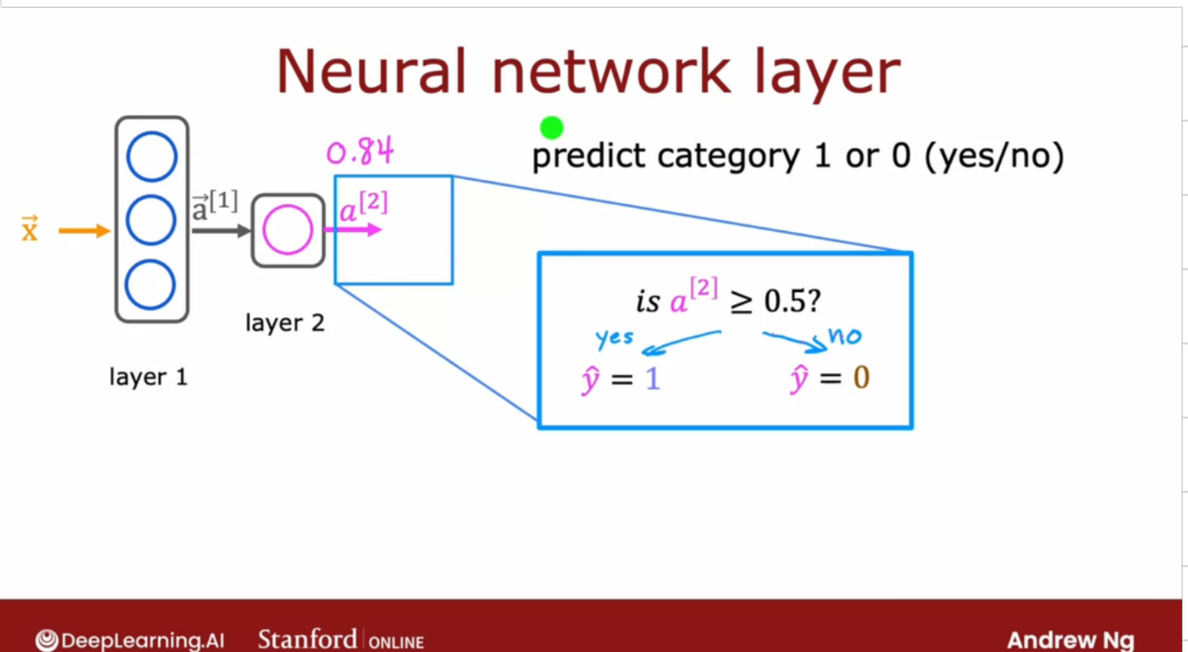
Day-27, Oct-29, 2024.

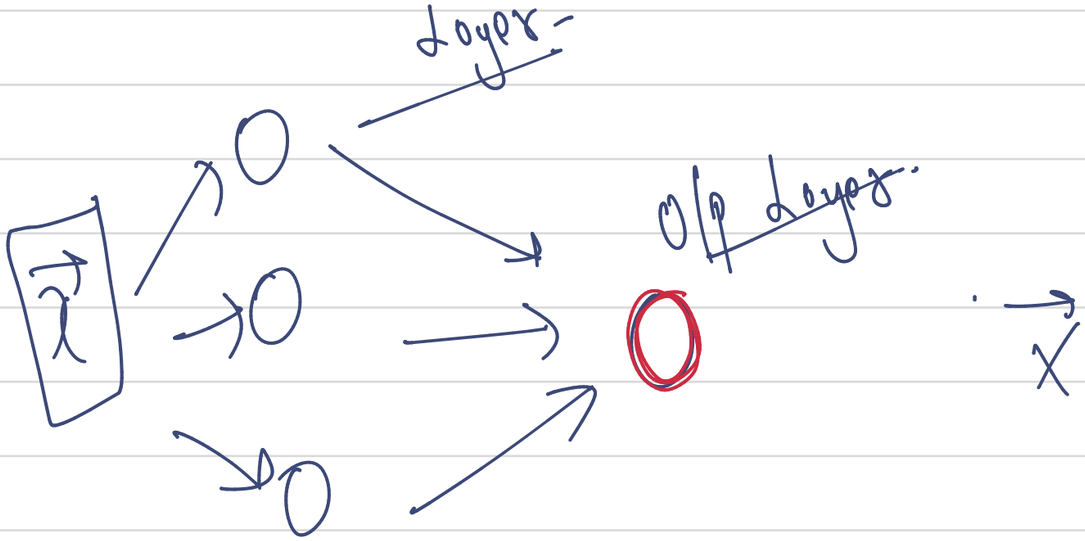




$\vec{x}$ is the input vector for the Input Layer where $\vec{x}$ can be $\begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$ $\xrightarrow{\vec{x}}$ Suppose first Input Layer → $\vec{w}_n, b_n$ goes to activation function







$\vec{x}$

$\vec{w}_1, b_1$

$a_1 = g(\vec{w}_1 \cdot \vec{x} + b_1) \rightarrow 0.3$

$\vec{w}_2, b_2$

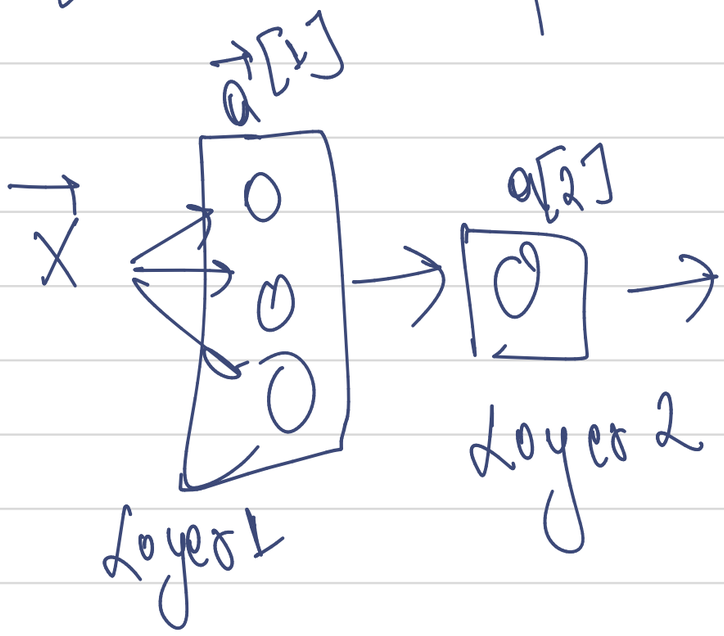
$a_2 = g(\vec{w}_2 \cdot \vec{x} + b_2) \rightarrow 0.7$

$\vec{w}_3, b_3$

$a_3 = g(\vec{w}_3 \cdot \vec{x} + b_3) \rightarrow 0.2$

Suppose

$\begin{bmatrix} 0.3 \\ 0.7 \\ 0.2 \end{bmatrix}$ are the output obtained from the Layer 1 to the final output Layer so -

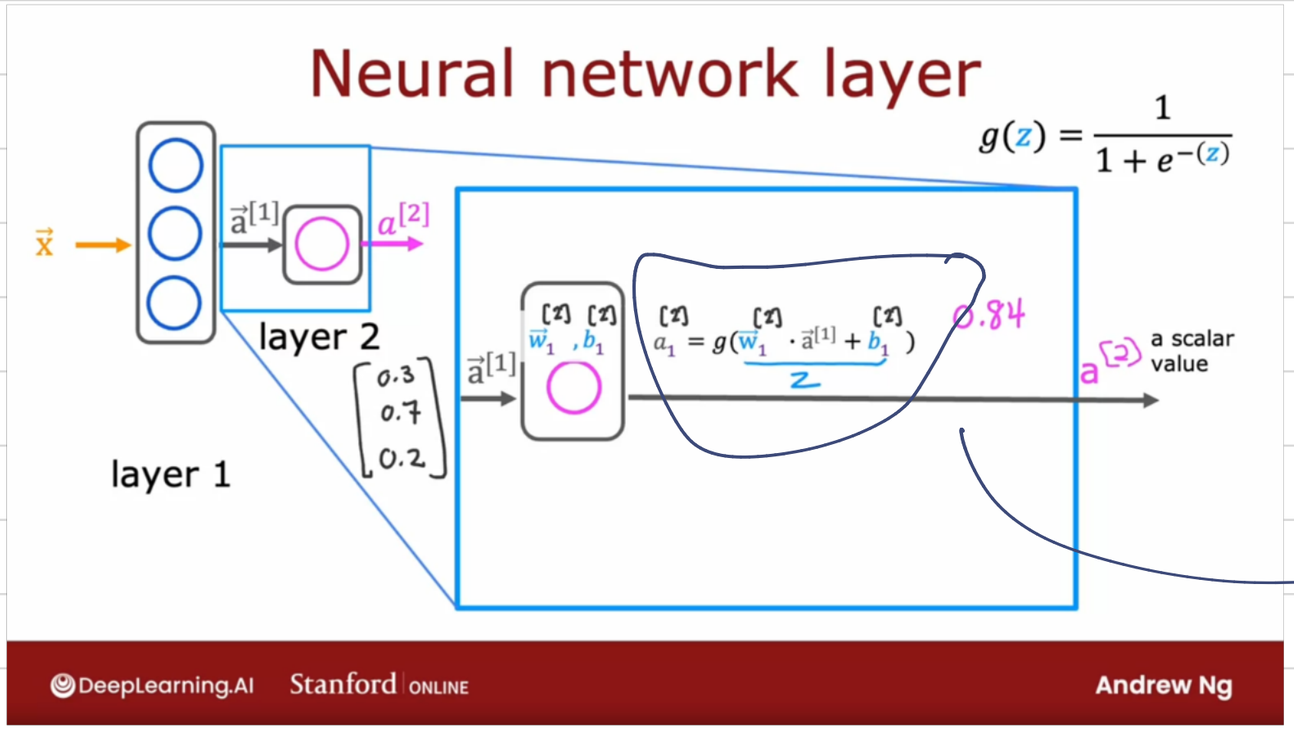


So, the Layer 2 has 0.7 then Spam Email ? Is $a^{[2]} \geq 0.5$

Yes Spam

No Spam

for this Case it is Spam Email since $a^{[2]}$ is 0.7 (Suppose).



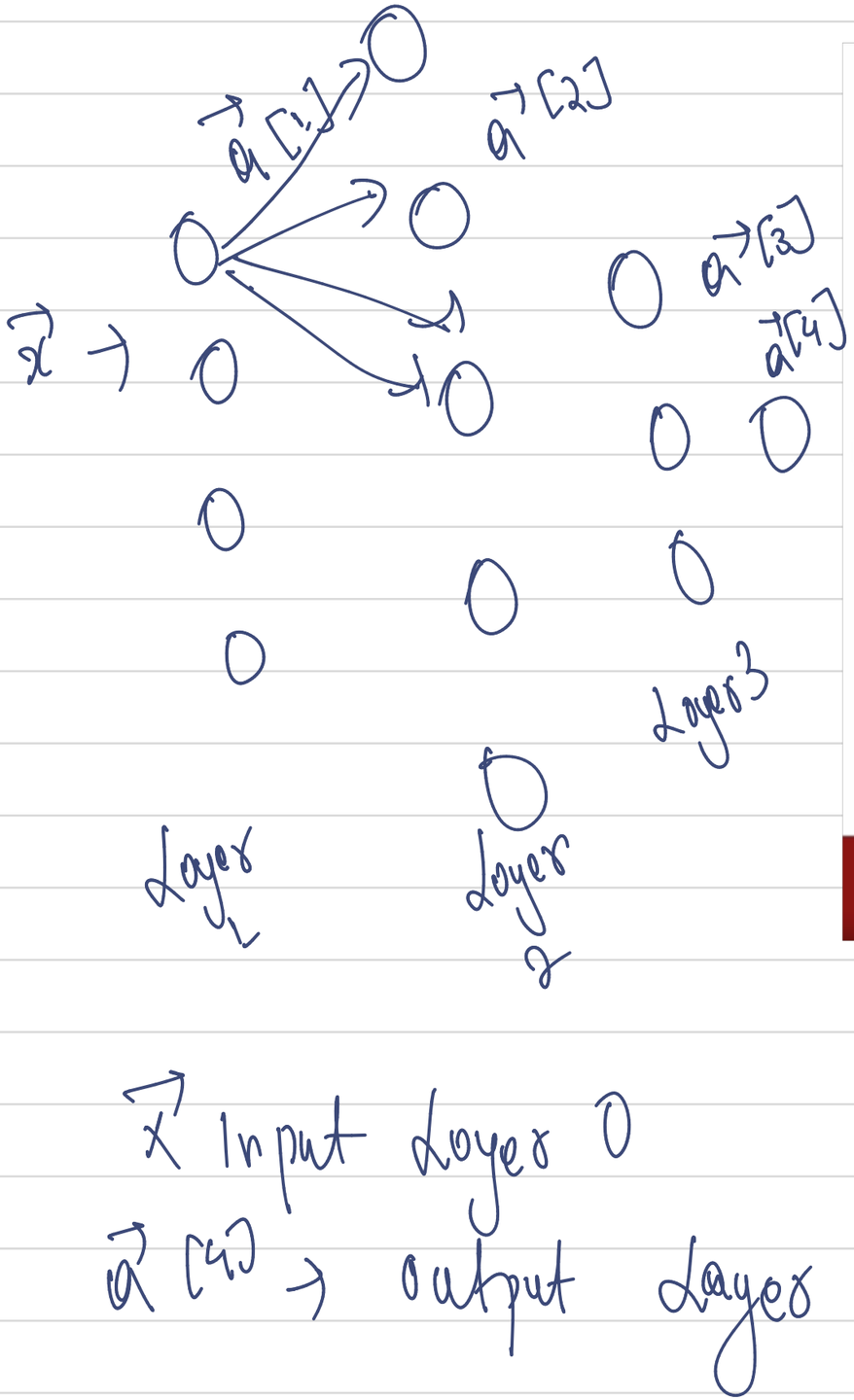
Dot product of two vectors gives → a scalar value.

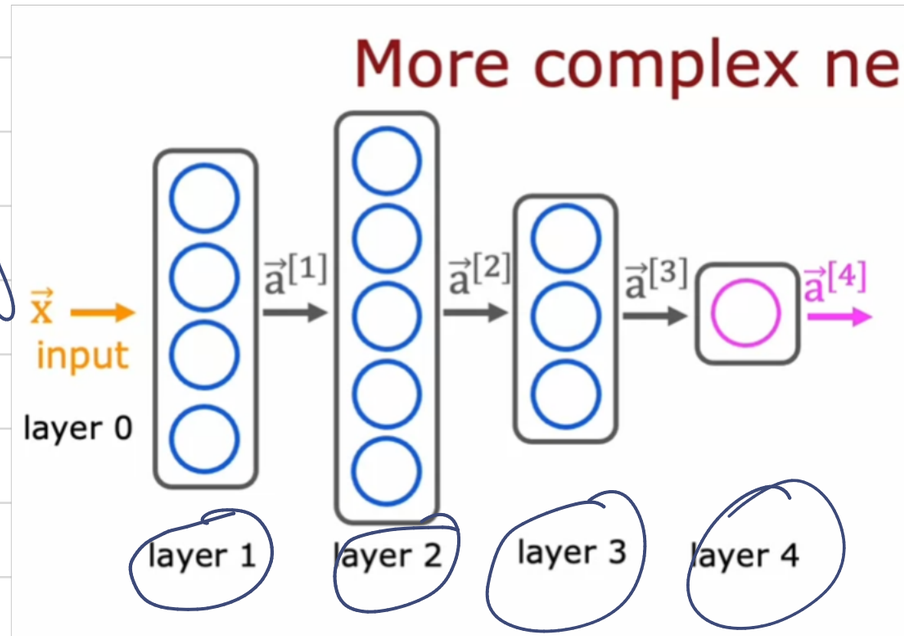
$$a_i = g(\vec{w}_i \cdot \vec{a}^{[i]} + b_i)$$

from the figure

So, for layer 1 $\vec{a}^{[1]}$ it is $\vec{w}_1, b_1$ and $\vec{a}^{[2]}$ $\vec{w}_2, b_2$ and so on. that depends upon the number of nodes you are using.

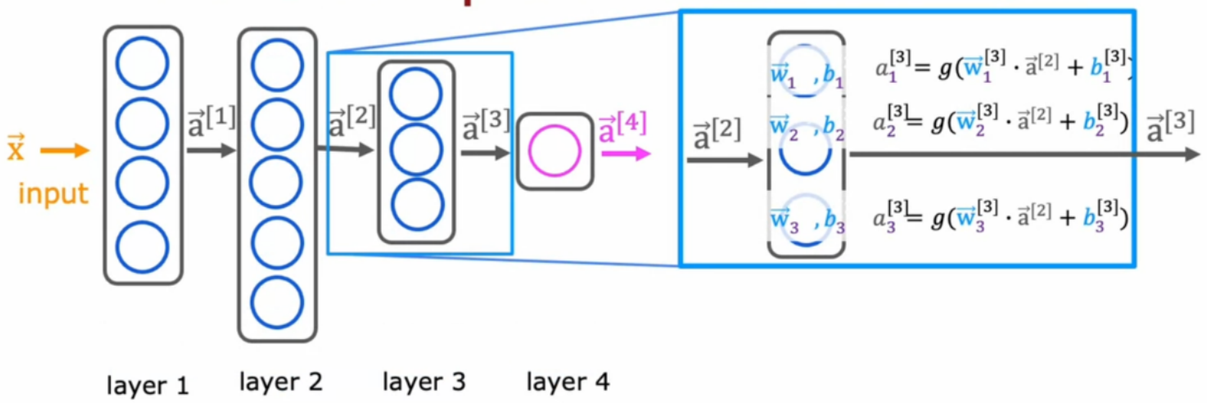
So it's a notation — for Complex Neural Networks we use more Complex Notation Subscript & Superscript.


$\vec{a}^{[1]}$
$\vec{a}^{[2]}$
$\vec{a}^{[3]}$
$\vec{a}^{[4]}$
$\vec{x}$
Layer 1
Layer 2
Layer 3
$\vec{x}$ Input Layer 0
$\vec{a}^{[4]}$ → output Layer

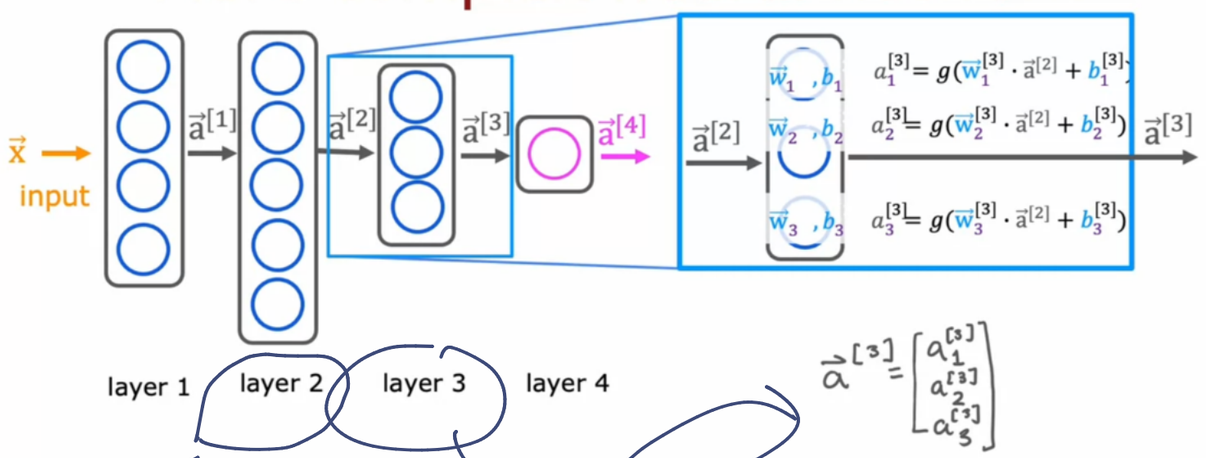

More complex neural network
$\vec{x}$
input
layer 0
$\vec{a}^{[1]}$
$\vec{a}^{[2]}$
$\vec{a}^{[3]}$
$\vec{a}^{[4]}$
layer 1
layer 2
layer 3
layer 4
DeepLearning.AI

More complex neural network
$\vec{x}$
input
layer 0
$\vec{a}^{[1]}$
$\vec{a}^{[2]}$
$\vec{a}^{[3]}$
$\vec{a}^{[4]}$
layer 1
layer 2
layer 3
layer 4
hidden layers
output layer
DeepLearning.AI
Stanford ONLINE
Andrew Ng

More complex neural network



More complex neural network



DeepLearning.AI Stanford ONLINE Andrew Ng

$\vec{a}^{[3]} \Rightarrow \begin{bmatrix} a_1^{[3]} \\ a_2^{[3]} \\ a_3^{[3]} \end{bmatrix}$ $\vec{a}^{[1]} \Rightarrow \begin{bmatrix} a_1^{[1]} \\ a_1^{[2]} \\ a_1^{[3]} \\ a_1^{[4]} \\ a_1^{[5]} \end{bmatrix}$ $\vec{a}^{[4]} \rightarrow \begin{bmatrix} a_1^{[4]} \end{bmatrix} \Rightarrow$ last/output layer.

So, $a_1^{[3]} \Rightarrow g(\vec{w}_1^{[3]} \cdot \vec{a}^{[2]} + b_1^{[3]})$

where $\vec{a}^{[2]}$ is previous layer Activation
$b_1^{[3]}$ → bias for 3 layer 1st node
$w_1^{[3]}$ → weight for 3 layer 1st node

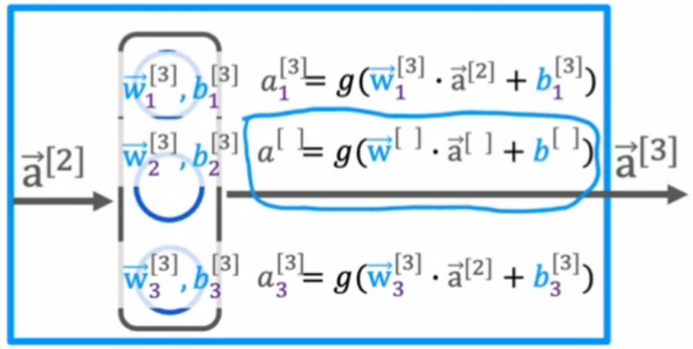
$\vec{a}^{[2]} \Rightarrow ?$

So for $\vec{a}^{[3]} \Rightarrow ?$

$\vec{a}^{[2]} \longrightarrow \vec{w}_3^{[3]}, b_3^{[3]} \Rightarrow$

$g(\vec{w}_3^{[3]} \cdot \vec{a}^{[2]} + b_3^{[3]})$

[In Video Quiz]



Can you fill in the superscripts and subscripts for the second neuron?

- ✓ $a_2^{[3]} = g(\vec{w}_2^{[3]} \cdot \vec{a}^{[2]} + b_2^{[3]})$
- $a_2^{[3]} = g(\vec{w}_2^{[3]} \cdot \vec{a}^{[3]} + b_2^{[3]})$
- $a_2^{[3]} = g(\vec{w}_2^{[3]} \cdot a_2^{[2]} + b_2^{[3]})$

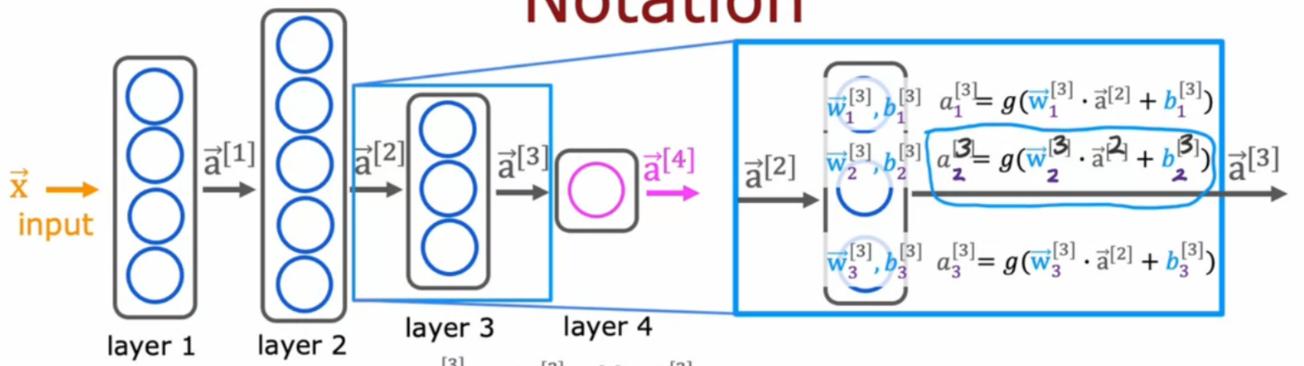
DeepLearning.AI Stanford ONLINE Andrew Ng

So, Mathematically, $\vec{a}^{[i]} \rightarrow w_j^{[K]}, b_j^{[K]} \rightarrow a^{[K]} = g(\vec{w}^{[K]} \cdot \vec{a}^{[i]} + b^{[K]} \rightarrow a^{[K]}$

for i = previous ith activation | $i \Rightarrow 4$

j = index of Kth layer $K \Rightarrow$ th layer for | $\vec{a}^{[4]} \rightarrow w_3^{[5]}, b_3^{[5]} \rightarrow a^{[5]} = g(\vec{w}^{[5]} \cdot \vec{a}^{[4]} + b^{[5]} \rightarrow a^{[5]}$

Notation



$a_2^{[3]} = g(\vec{w}_2^{[3]} \cdot \vec{a}^{[2]} + b_2^{[3]})$

Activation value of layer l, unit(neuron) j

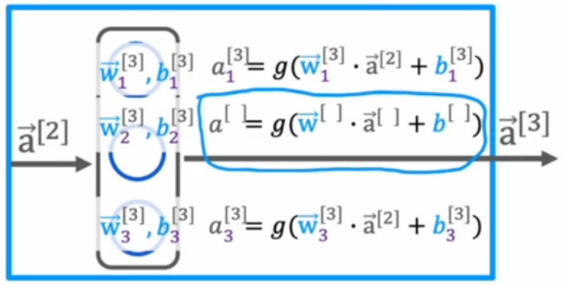
output of layer $l-1$ (previous layer)

$$a_j^{[l]} = g(\vec{w}_j^{[l]} \cdot \vec{a}^{[l-1]} + b_j^{[l]})$$

sigmoid "activation function"

Parameters w & b of layer l, unit j

[In Video Quiz]



Can you fill in the superscripts and subscripts for the second neuron?

- ✓ $a_2^{[3]} = g(\vec{w}_2^{[3]} \cdot \vec{a}^{[2]} + b_2^{[3]})$
- $a_2^{[3]} = g(\vec{w}_2^{[3]} \cdot \vec{a}^{[3]} + b_2^{[3]})$
- $a_2^{[3]} = g(\vec{w}_2^{[3]} \cdot a_2^{[2]} + b_2^{[3]})$

Formula for Activation function —

$$a_j^{[l]} = g(\vec{w}_j^{[l]} \cdot \vec{a}^{[l-1]} + b_j^{[l]})$$

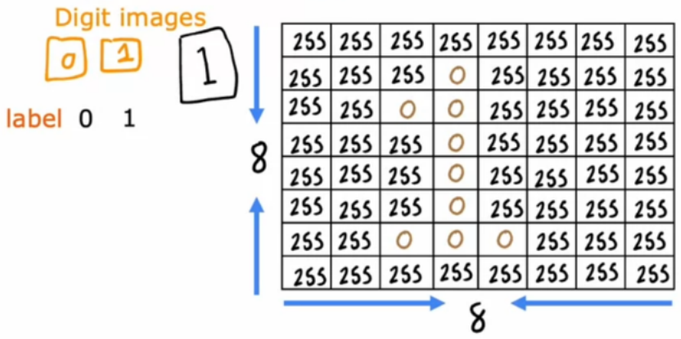
where $l \rightarrow$ layer

Unit $j \rightarrow$ of layer unit

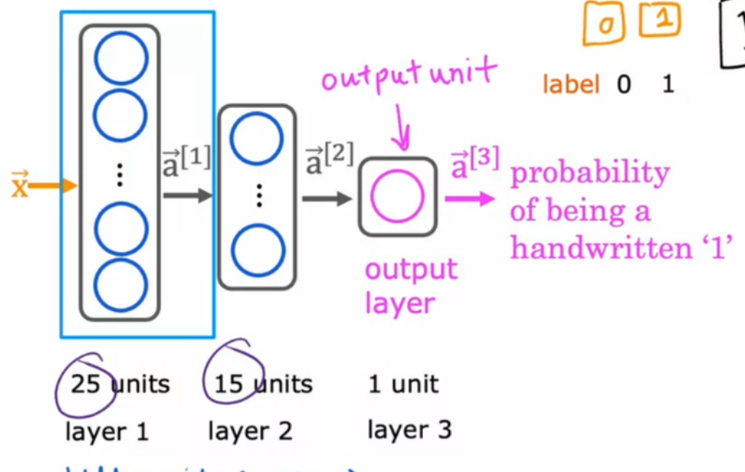
$\vec{a}^{[l-1]}$ previous activation

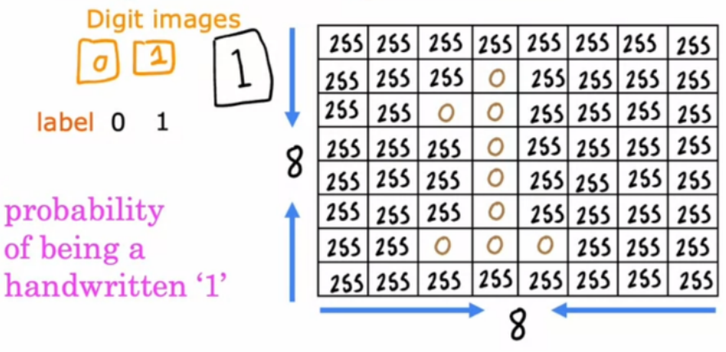
$\vec{a}_3^{[3]}$ is the Activation of layer 3. $\vec{a}_3^{[3]}$ followed $\vec{a}^{[2]}$ which is the activated function from layer 2.

Handwritten digit recognition



Handwritten digit recognition





DeepLearning.AI Stanford | ONLINE Andrew Ng

We have used 8x8 matrix for Image, so 3x8x8 where 3 is RGB

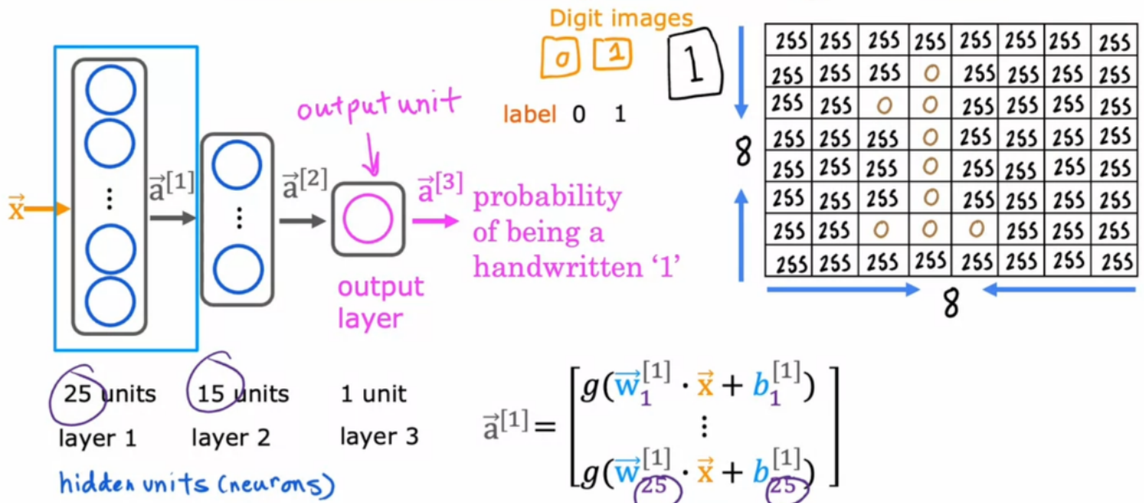
Input Layer → $\vec{x}$

Layer 1 → 25 units/neurons $(\vec{a}^{[1]})$

Layer 2 → 15 units/neurons $(\vec{a}^{[2]})$

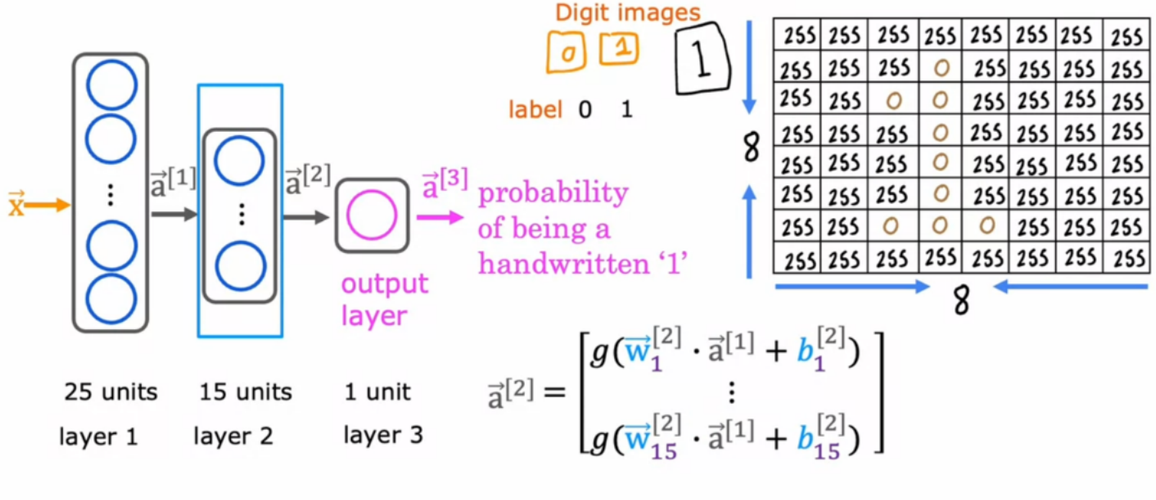
Layer 3 → 1 unit / 1 output Neuron $(\vec{a}^{[3]})$ that gives probability between [0, 1]

Handwritten digit recognition



$$\vec{a}^{[1]} = \begin{bmatrix} g(\vec{w}_1^{[1]} \cdot \vec{x} + b_1^{[1]}) \\ \vdots \\ g(\vec{w}_{25}^{[1]} \cdot \vec{x} + b_{25}^{[1]}) \end{bmatrix}$$

Handwritten digit recognition



$$\vec{a}^{[2]} = \begin{bmatrix} g(\vec{w}_1^{[2]} \cdot \vec{a}^{[1]} + b_1^{[2]}) \\ \vdots \\ g(\vec{w}_{15}^{[2]} \cdot \vec{a}^{[1]} + b_{15}^{[2]}) \end{bmatrix}$$

DeepLearning.AI Stanford ONLINE Andrew Ng

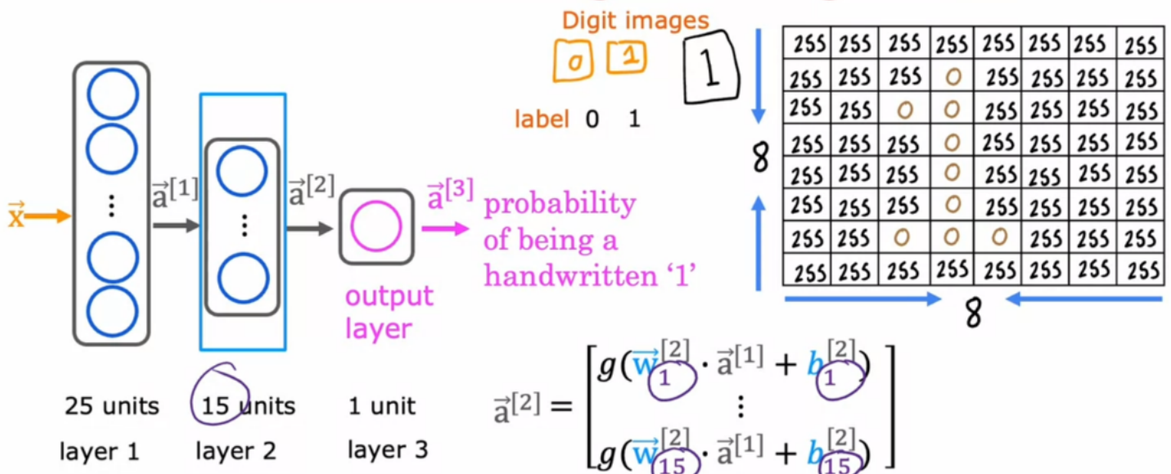
Let's say we have 30 units then we'll have 30 bias → Layer 1

$$\vec{a}^{[1]} \Rightarrow \begin{bmatrix} g(\vec{w}_1^{[1]} \cdot \vec{x} + b_1^{[1]}) \\ \vdots \\ g(\vec{w}_{30}^{[1]} \cdot \vec{x} + b_{25}^{[1]}) \end{bmatrix}$$

So, $\vec{a}^{[l]} \Rightarrow \begin{bmatrix} g(\vec{w}_i^{[l]} \cdot \vec{x} + b_i^{[l]}) \\ \vdots \\ g(\vec{w}_i^{[l]} \cdot \vec{x} + b_i^{[l]}) \end{bmatrix}$

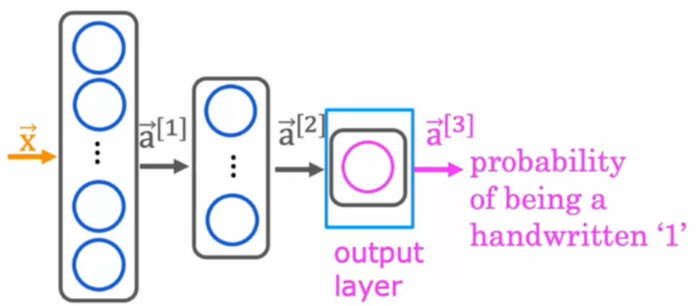
for ith term.
l = Layer.

Handwritten digit recognition



$$\vec{a}^{[2]} = \begin{bmatrix} g(\vec{w}_1^{[2]} \cdot \vec{a}^{[1]} + b_1^{[2]}) \\ \vdots \\ g(\vec{w}_{15}^{[2]} \cdot \vec{a}^{[1]} + b_{15}^{[2]}) \end{bmatrix}$$

Handwritten digit recognition



$$\vec{a}^{[3]} = \left[g\left(\vec{w}_1^{[3]} \cdot \vec{a}^{[2]} + b_1^{[3]} \right) \right]$$

is $a_1^{[3]} \geq 0.5$?

yes → $\hat{y} = 1$ image is digit 1

no → $\hat{y} = 0$ image isn't digit 1

$$\vec{a}^{[3]} \Rightarrow \left[g\left(\vec{w}_1^{[3]} \cdot \vec{a}^{[2]} + b_1^{[3]} \right) \right]$$

is $a_1^{[3]} \geq 0.5$?

Yes → $\hat{y} \Rightarrow 1$ Image digit 1

No → $\hat{y} \Rightarrow 0$ Image isn't digit 1

So, if $a_1^{[3]} \Rightarrow 0.85$

it is digit '1'

otherwise it is '0'