

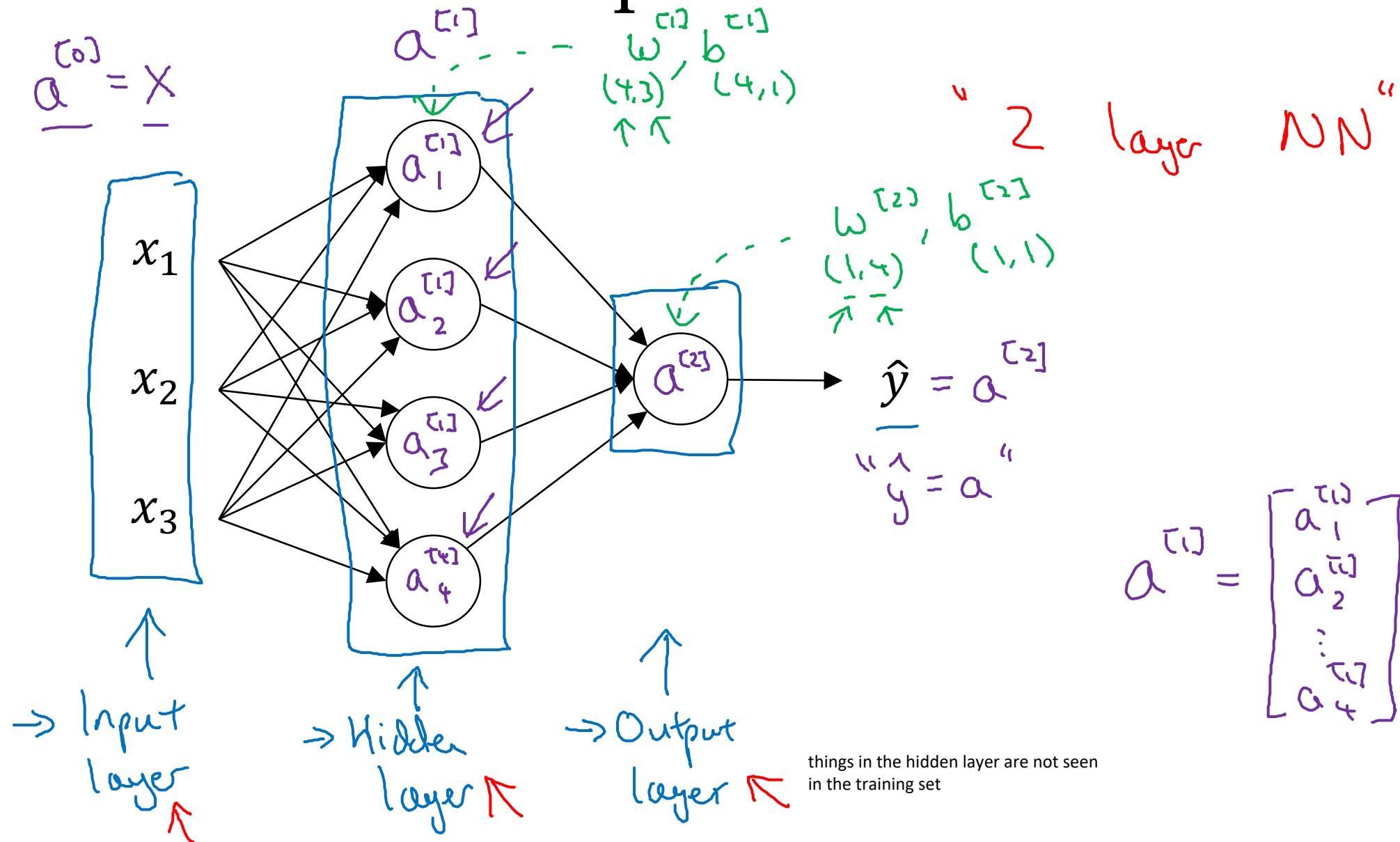


deeplearning.ai

One hidden layer
Neural Network

Neural Network
Representation

Neural Network Representation



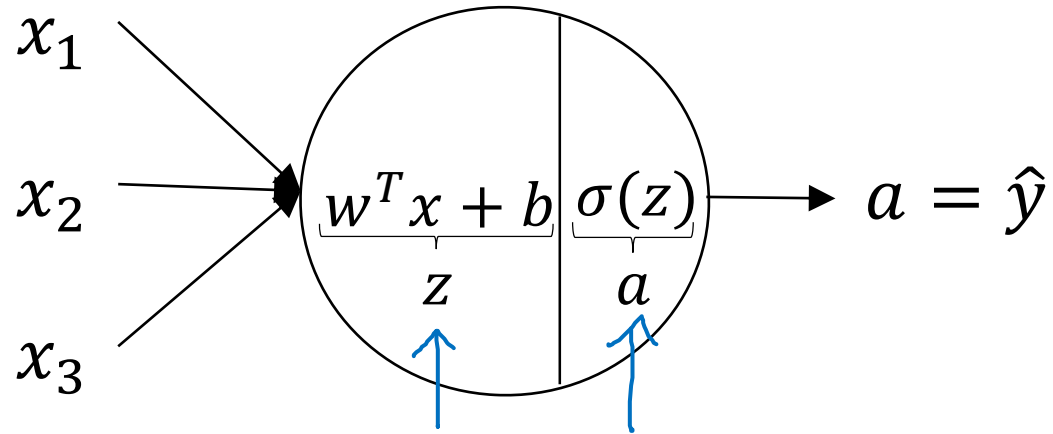


deeplearning.ai

One hidden layer Neural Network

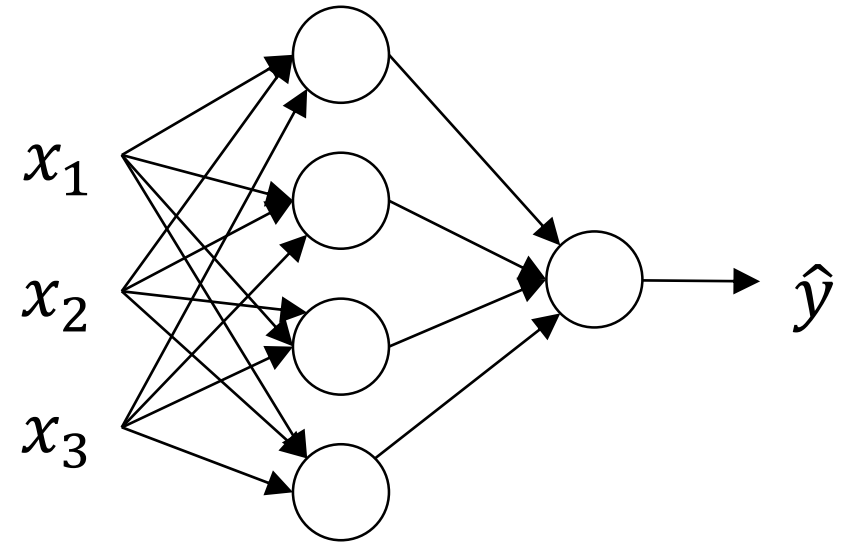
Computing a
Neural Network's
Output

Neural Network Representation

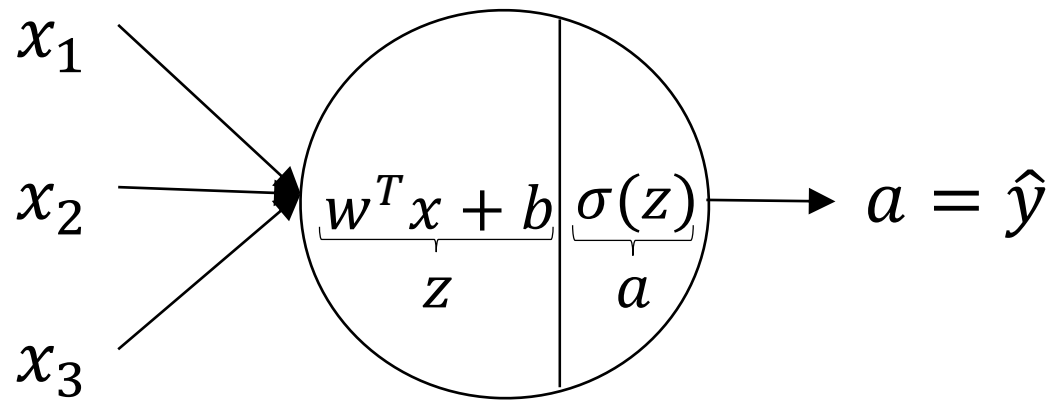


$$z = w^T x + b$$

$$a = \sigma(z)$$

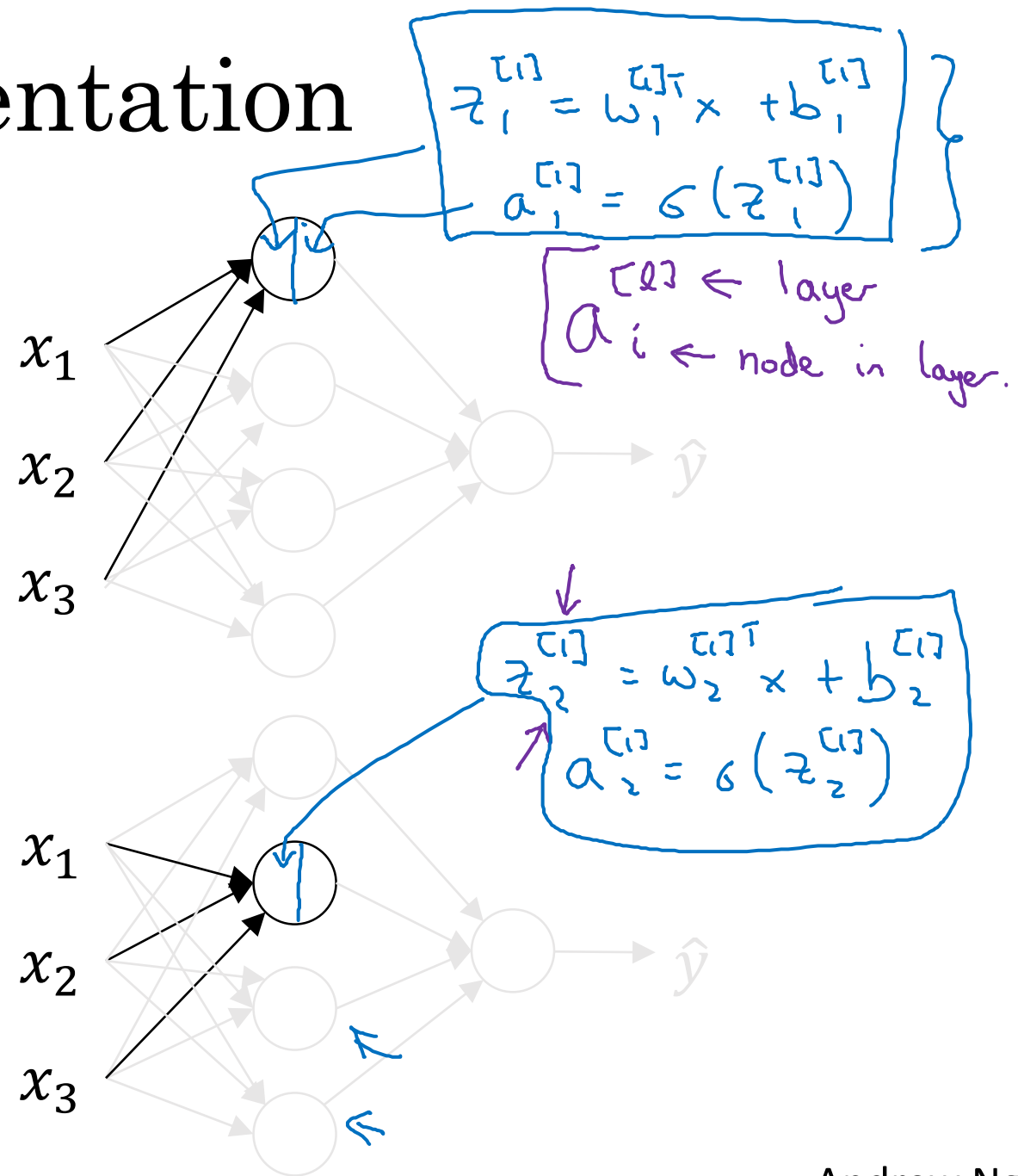


Neural Network Representation

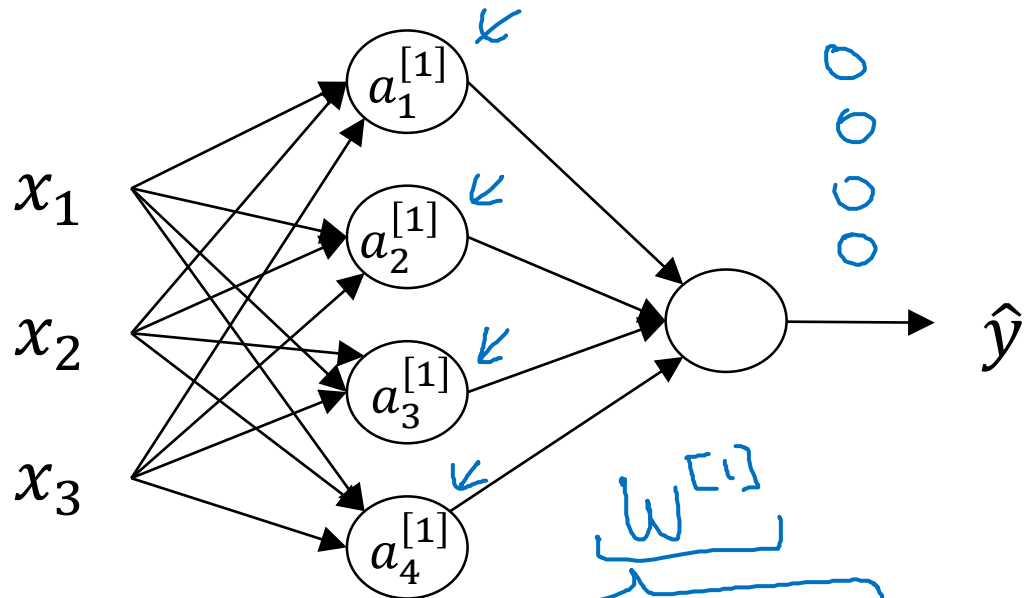


$$z = w^T x + b$$

$$a = \sigma(z)$$



Neural Network Representation



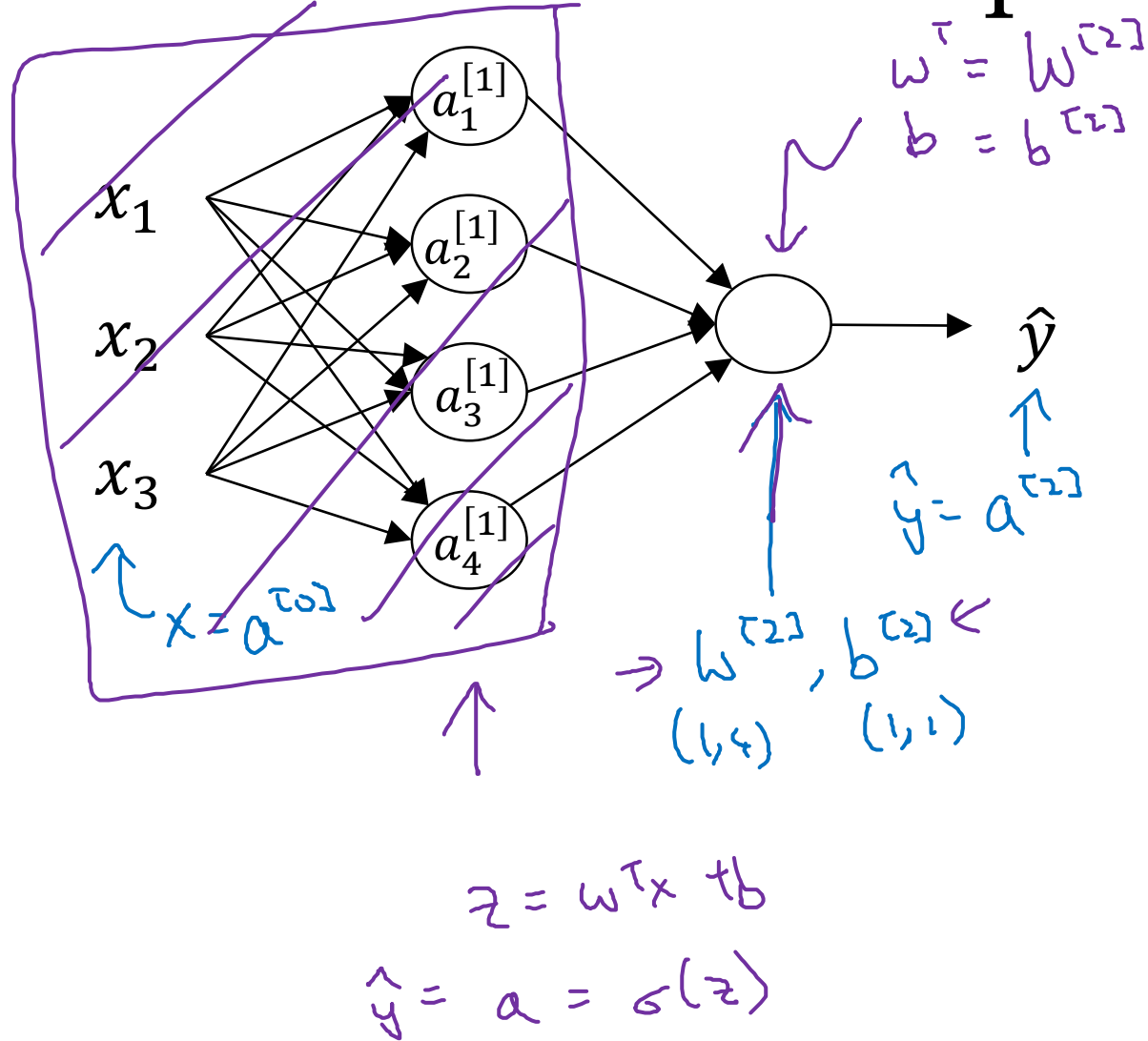
$$\begin{aligned}
 z_1^{[1]} &= w_1^{[1]T} x + b_1^{[1]} & a_1^{[1]} &= \sigma(z_1^{[1]}) \\
 z_2^{[1]} &= w_2^{[1]T} x + b_2^{[1]} & a_2^{[1]} &= \sigma(z_2^{[1]}) \\
 z_3^{[1]} &= w_3^{[1]T} x + b_3^{[1]} & a_3^{[1]} &= \sigma(z_3^{[1]}) \\
 z_4^{[1]} &= w_4^{[1]T} x + b_4^{[1]} & a_4^{[1]} &= \sigma(z_4^{[1]})
 \end{aligned}$$

Handwritten notes: $(w_1^{[1]})^T x$ and $Q^{[1]}$ are written above the first equation. A red box highlights the activation function part $a_i^{[1]} = \sigma(z_i^{[1]})$. A blue arrow points from the hidden layer nodes to the equations.

$$\begin{aligned}
 &\rightarrow z^{[1]} = \begin{bmatrix} -w_1^{[1]T} \\ -w_2^{[1]T} \\ -w_3^{[1]T} \\ -w_4^{[1]T} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix} = \begin{bmatrix} \rightarrow w_1^{[1]T} x + b_1^{[1]} \\ \rightarrow w_2^{[1]T} x + b_2^{[1]} \\ \rightarrow w_3^{[1]T} x + b_3^{[1]} \\ \rightarrow w_4^{[1]T} x + b_4^{[1]} \end{bmatrix} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix} \\
 &\rightarrow a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ \vdots \\ a_4^{[1]} \end{bmatrix} = \sigma(z^{[1]})
 \end{aligned}$$

Handwritten notes: $(4, 3)$ is written below the weight matrix. $b^{[1]} (4, 1)$ is written below the bias vector. A red box highlights the activation function part $a^{[1]} = \sigma(z^{[1]})$. A blue arrow points from the hidden layer nodes to the equations.

Neural Network Representation learning



Given input x :

$$\begin{aligned}
 \rightarrow z^{[1]} &= W^{[1]} a^{[0]} + b^{[1]} \\
 &\quad (4,1) \quad (4,3) \quad (3,1) \quad (4,1) \\
 \rightarrow a^{[1]} &= \sigma(z^{[1]}) \\
 &\quad (4,1) \quad (4,1) \\
 \rightarrow z^{[2]} &= W^{[2]} a^{[1]} + b^{[2]} \\
 &\quad (1,1) \quad (1,4) \quad (4,1) \quad (1,1) \\
 \rightarrow a^{[2]} &= \sigma(z^{[2]}) \\
 &\quad (1,1) \quad (1,1)
 \end{aligned}$$

given a single input feature vector x , you can with four lines of code compute the outputs of this neural network