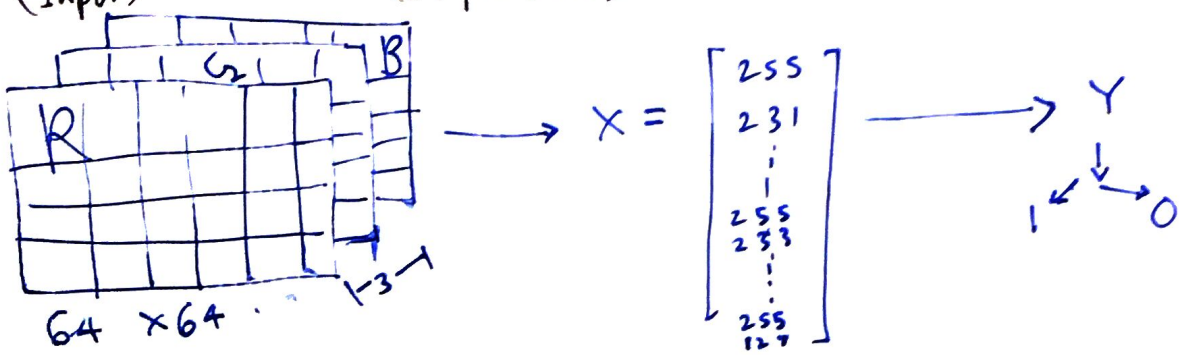
Deep learning

Binary Classification Problems:

eg Image (Input) ⟶ label: 0 or 1 (output label)



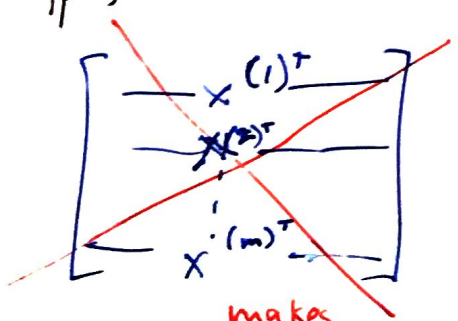
$64 \times 64 \times 3 = 12288$ $\qquad$ 12288

notation:

assuming m training examples, $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$

m_{test} = # test examples = no. of test examples

$$X = \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ | & | & & | \end{bmatrix} \quad n_x \text{ (rows)}, \quad m \text{ (columns)}$$



makes neural network coding HARDER

$X.shape = (n_x, m)$

Similarly

$$Y = \begin{bmatrix} y^{(1)}, & y^{(2)} & \cdots, & y^{(m)} \end{bmatrix}$$

Logistic Regression:

⟶ learning algorithm used when output labels (y), in a supervised learning problem, are either 0 or 1 [i.e. for Bin. Class. Probs]

Given x, want $\hat{y} = P(y=1 \mid x)$

$x \in \mathbb{R}^{n_x}$

Parameters: $w \in \mathbb{R}^{n_x}$, $b \in \mathbb{R}$