



deeplearning.ai

Basics of Neural Network Programming

Binary Classification

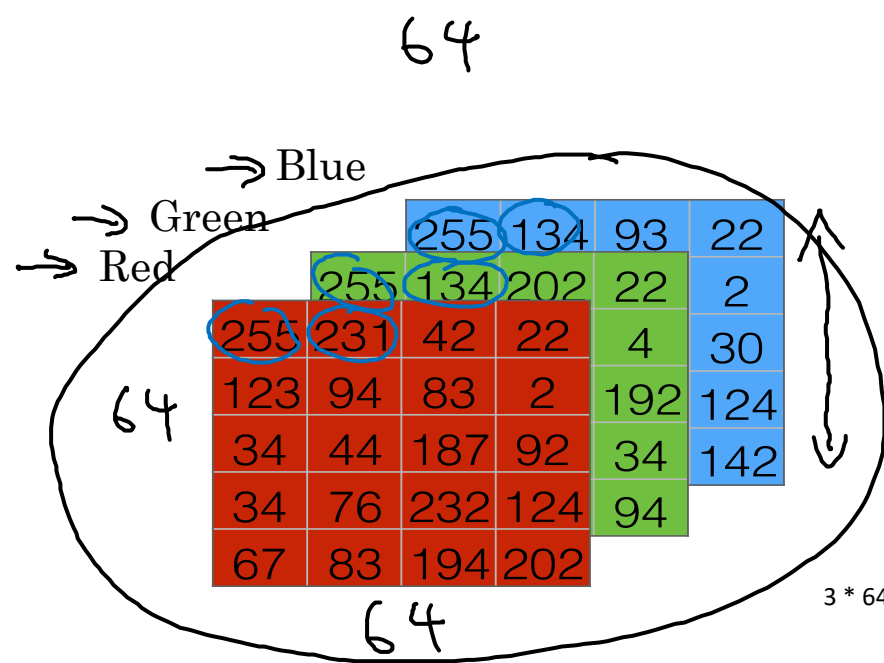
logistics regression is an algorithm for binary classification.

Binary Classification

So in binary classification, our goal is to learn a classifier that can input an image represented by feature vector x . And predict whether the corresponding label y is 1 or 0.



→ 1 (cat) vs 0 (non cat)
 y



$x =$

255
231
⋮
⋮
⋮
255
134
⋮

$64 \times 64 \times 3 = 12288$

$n = n_x = 12288$ dimension of this input feature vector

$x \rightarrow y$

Notation

$$(x, y) \quad x \in \mathbb{R}^{n_x}, y \in \{0, 1\} \quad n_x = 12288$$

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

$$M = M_{\text{train}}$$

$M_{\text{test}} = \# \text{test examples.}$

The diagram illustrates the matrix X and its dimensions. On the left, a large matrix X is shown with columns labeled $x^{(1)}, x^{(2)}, \dots, x^{(m)}$. A vertical double-headed arrow on the right indicates the height is n_x . A horizontal double-headed arrow at the bottom indicates the width is m . To the right of this matrix is a smaller square matrix with a large 'X' drawn over it, indicating it is not the convention. This crossed-out matrix has labels $x^{(1)}$ and $x^{(m)}$ on its columns, and a vertical double-headed arrow on its right indicating height m .

$$X \in \mathbb{R}^{n_x \times m}$$

$$X.\text{shape} = (n_x, m)$$

$$Y = [y^{(1)} \ y^{(2)} \ \dots \ y^{(m)}]$$

$$Y \in \mathbb{R}^{1 \times m}$$

$$Y.\text{shape} = (1, m)$$

It turns out that when you're implementing neural networks using this convention I have on the left will make the implementation much easier.