deeplearning.ai

Basics of Neural Network Programming

Binary Classification

logistics regression is an algorithm for binary classification.

Binary Classification

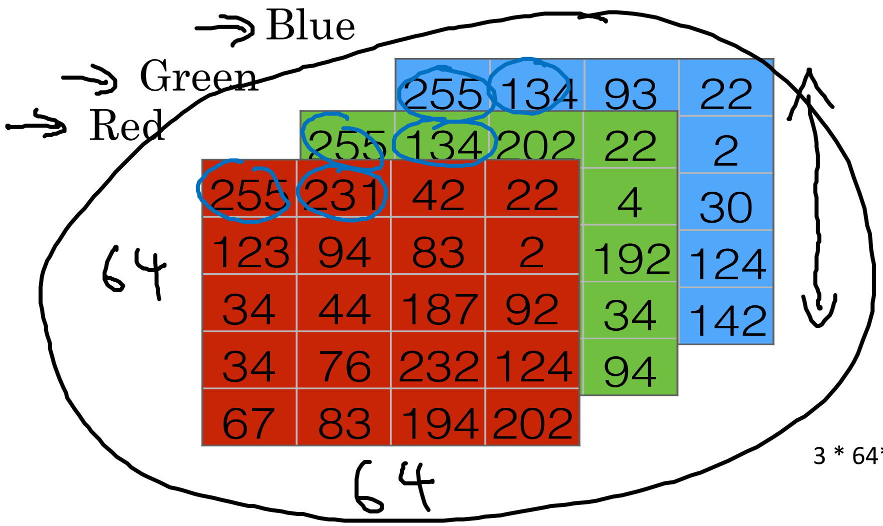
So in binary classification, our goal is to learn a classifier that can input an image represented by feature vector x. And predict whether the corresponding label y is 1 or 0.

64

64

1 (cat) vs 0 (non cat)

y



3 * 64*64 matrices

$$x = \begin{bmatrix} 255 \\ 231 \\ \vdots \\ \vdots \\ 255 \\ 134 \\ \vdots \end{bmatrix}$$

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

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

$$x \longrightarrow y$$

Andrew Ng

Notation

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

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

$M = M_{train}$ $\qquad M_{test}$ = #test examples.

$$X = \underbrace{\begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ | & | & & | \end{bmatrix}}_{m} \Big\updownarrow n_x \qquad \cancel{\begin{bmatrix} - & x^{(1)T} & - \\ & \vdots & \\ - & x^{(m)T} & - \end{bmatrix}}$$

$X \in \mathbb{R}^{n_x \times m}$ $\qquad X.\text{shape} = (n_x, m)$

$Y = [y^{(1)} \; y^{(2)} \; \ldots, y^{(m)}]$

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

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

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