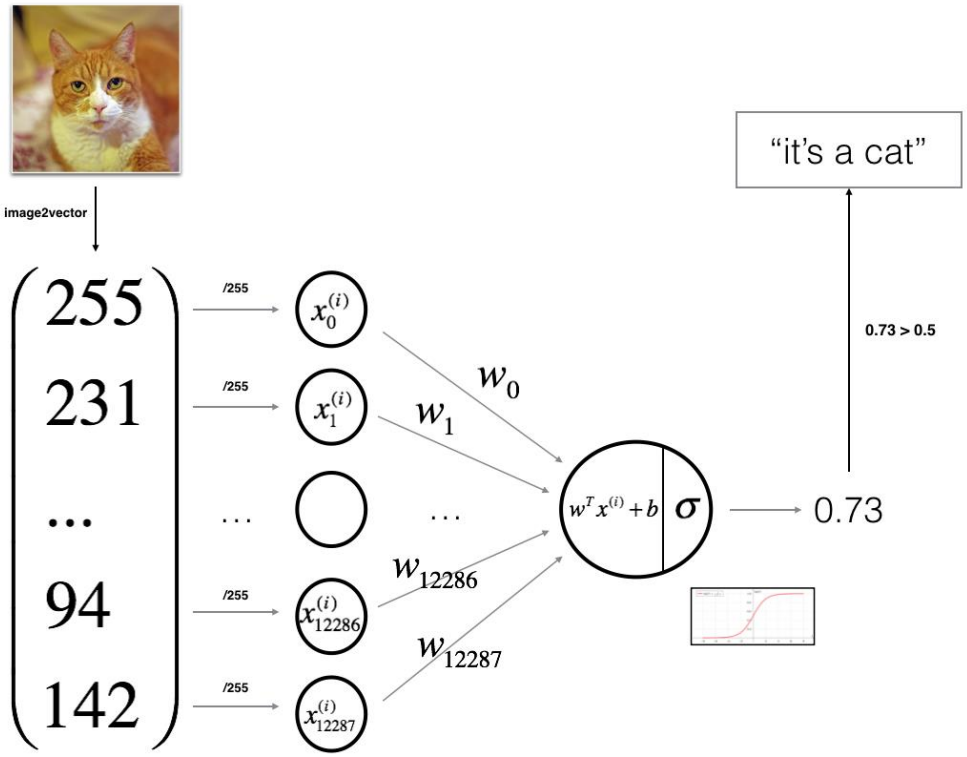
Logistic Regression model (with a Neural Network mindset) for week 3

The model:



The general methodology:

The main steps for building the model:

1) Define the model structure (such as number of input features)
2) Initialize the model's parameters
3) Loop:
   - Calculate current loss (forward propagation)
   - Calculate current gradient (backward propagation)
   - Update parameters (gradient descent)

Helper functions compute steps 1-3, and then merged into one function (called *model*)

Mathematical expression of the algorithm:

Forward propagation:

For one training example $x^{(i)}$:

$$z^{(i)} = w^T x^{(i)} + b$$
$$\hat{y}^{(i)} = a^{(i)} = \sigma(z^{(i)})$$
$$L(a^{(i)}, y^{(i)}) = -y^{(i)} \log(a^{(i)}) - (1 - y^{(i)}) \log(1 - a^{(i)})$$

The cost is computed by summing the loss $\left(L(a^{(i)}, y^{(i)})\right)$ over all training examples:

$$J = \frac{1}{m} \sum_{i=1}^{m} L(a^{(i)}, y^{(i)})$$

Backward propagation:

$$\frac{\partial J}{\partial w} = \frac{1}{m} X(A - Y)^T$$
$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} \left(a^{(i)} - y^{(i)}\right)$$

$NOTE$: $\frac{\partial J}{\partial Z} = A - Y$

Update parameters:

$$w := w - training\ rate * \frac{\partial J}{\partial w}$$
$$b := b - training\ rate * \frac{\partial J}{\partial b}$$

Predictions:

Calculate:

$$\hat{Y} = A = \sigma(w^T X + b)$$

Convert the entries of a into 0 (if activation ≤ 0.5) or 1 (if activation > 0.5):

$$y_{prediction} = \begin{cases} 1 & activation > 0.5 \\ 0 & otherwise \end{cases}$$