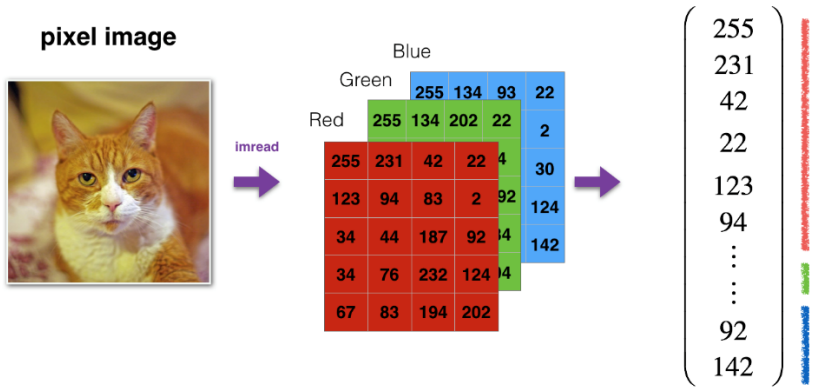
Deep Neural Network for Image Classification

Dataset

- A training set of *m_train* images labelled as cat (1) or non-cat (0)
- a test set of *m_test* images labelled as cat and non-cat
- each image is of shape (num_px, num_px, 3) where 3 is for the 3 channels (RGB).

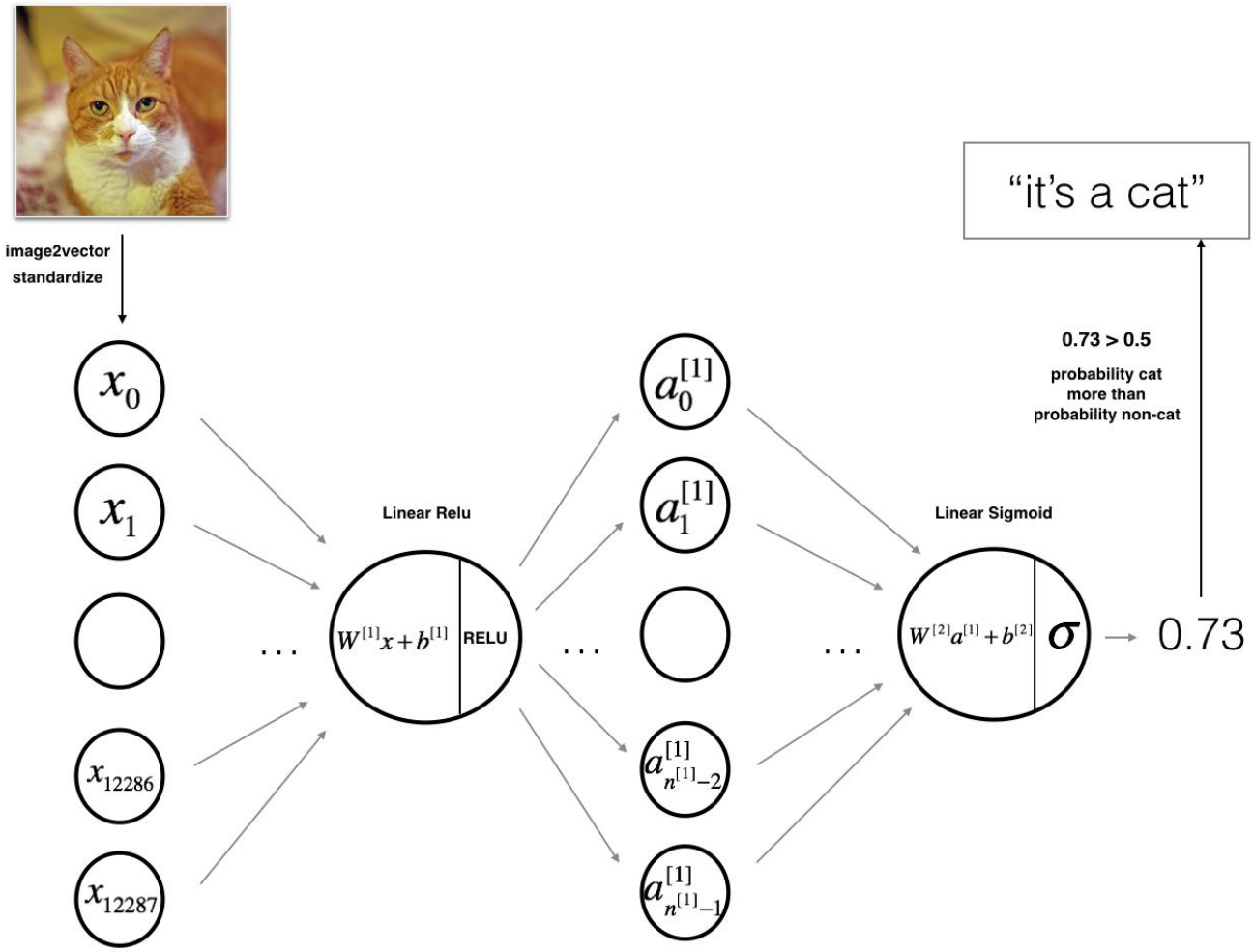
reshape and standardize the images before feeding them to the network:



Architecture of the model

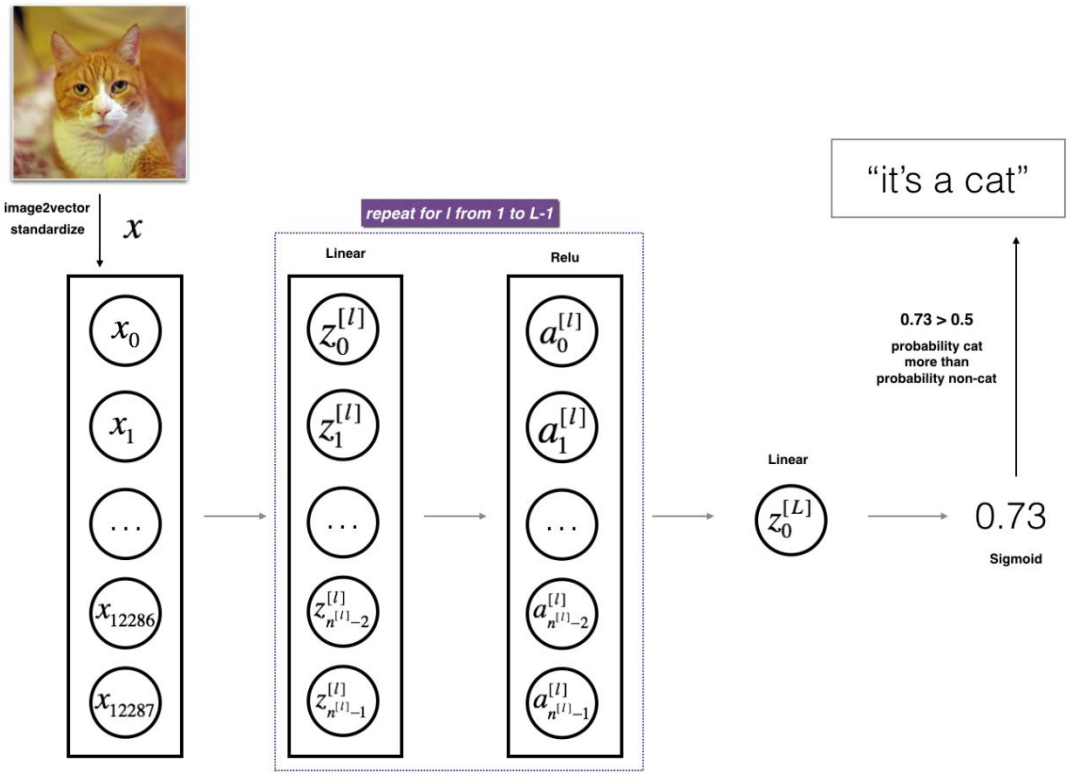
We build two different models:

A 2-layer neural network: *two_layer_model*



- The input is a (64,64,3) image which is flattened to a vector of size (12288,1).
- The corresponding vector: $[x_0, x_1, \dots, x_{12287}]^T$ is then multiplied by the weight matrix $W^{[1]}$ of size $(n^{[1]}, 12288)$.
- You then add a bias term and take its ReLU to get the following vector: $\left[a_0^{[1]}, a_1^{[1]}, \dots, a_{n^{[1]}-1}^{[1]}\right]^T$
- You then repeat the same process.
- You multiply the resulting vector by $W^{[2]}$ and add your intercept (bias).
- Finally, you take the sigmoid of the result. If it is greater than 0.5, you classify it to be a cat.

An L-layer neural network:



- The input is a (64,64,3) image which is flattened to a vector of size (12288,1).
- The corresponding vector: $[x_0, x_1, \dots, x_{12287}]^T$ is then multiplied by the weight matrix $W^{[1]}$ and then you add the intercept $b^{[1]}$. The result is called the linear unit.
- Next, you take the ReLU of the linear unit. This process could be repeated several times for each $(W^{[l]}, b^{[l]})$ depending on the model architecture.
- Finally, you take the sigmoid of the final linear unit. If it is greater than 0.5, you classify it to be a cat.

General methodology

Deep Learning methodology to build the model:

1) Initialize parameters / Define hyperparameters
2) Loop for *num_iterations*:
    a) Forward propagation
    b) Compute cost function
    c) Backward propagation
    d) Update parameters (using parameters, and grads from backprop)
3) Use trained parameters to predict labels

All functions used are described in 'Building your deep neural network step by step'