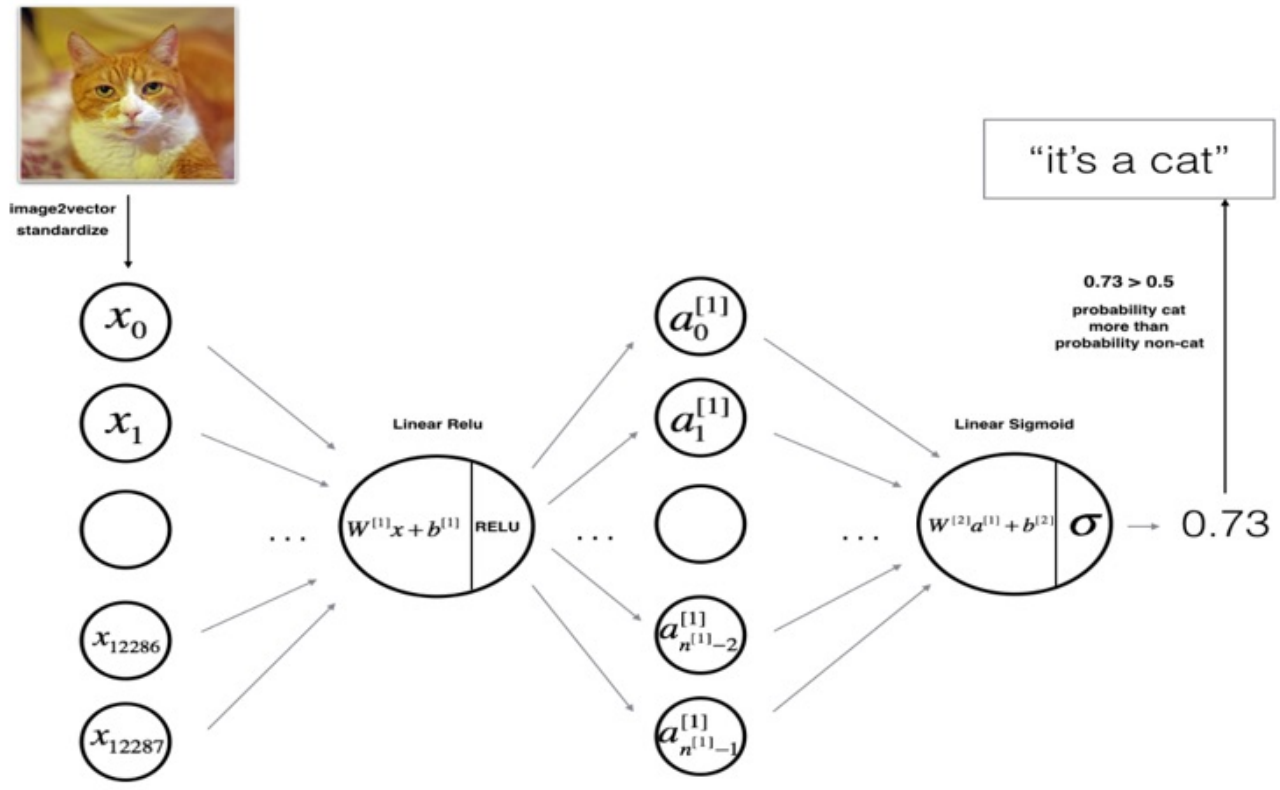
Deep Neural Network for Image Classification

1. 2-Layer Neural Network



- The input is a (64,64,3) image which is flattened to a vector of size $(12288, 1)$.
- The corresponding vector: $[x_0, x_1, ..., x_{12287}]^T$ 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: $[a_0^{[1]}, a_1^{[1]}, ..., a_{n^{[1]}-1}^{[1]}]^T$.
- 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.

(1) Initialize parameters

parameters = initialize_parameters(n_x, n_h, n_y)

↓ ↳ layer_dims

W1 / b1 / W2 / b2

(2) Loop (gradient descent)

for i in range(0, num_iterations):

1. forward propagation

A1, cache = linear_activation_forward(X, W1, b1, activation="relu")

A2, cache = linear_activation_forward(A1, W2, b2, activation="sigmoid")

↓ ↓

LINEAR → RELU → LINEAR → SIGMOID

ii. Compute cost

```
cost = compute_cost (A2, Y)
```

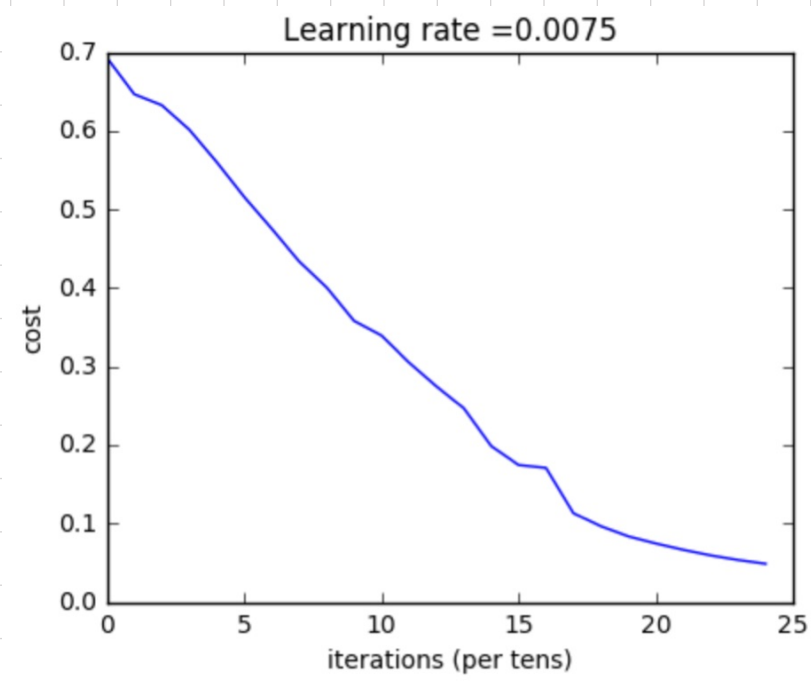
iii. Backward propagation

```
dA1, dW2, db2 = linear_activation_backward(dA2, cache2, activation="sigmoid")
dA0, dW1, db1 = linear_activation_backward(dA1, cache1, activation="relu")
grads['dW1'] = dW1
...
```

iv. update parameters

```
parameters = update_parameters(parameters, grads, learning_rate = learning_rate)
W1 = parameters['W1']
...
```

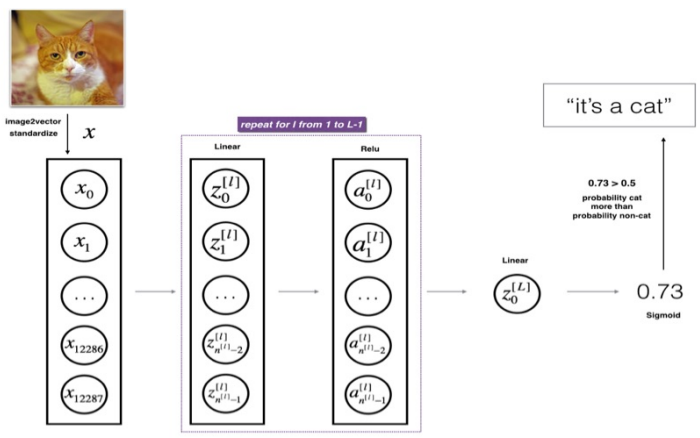
(3) output



train: 100%

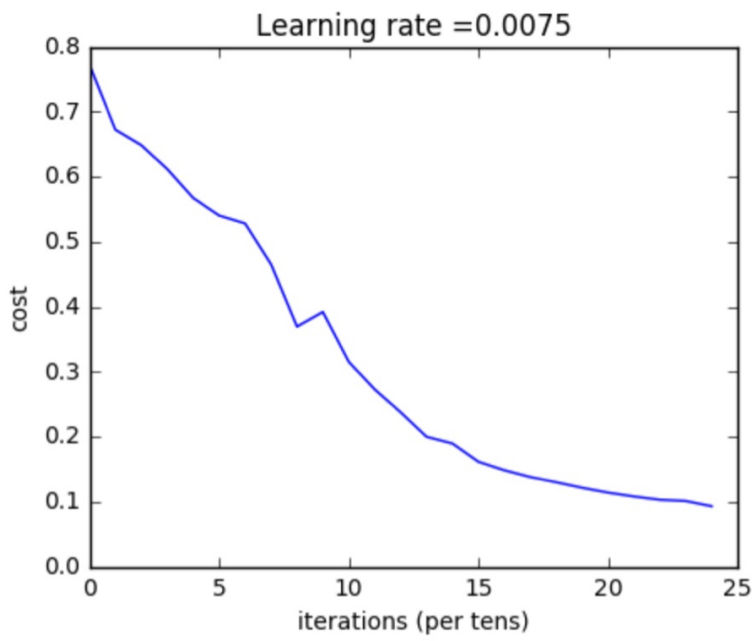
test: 72%

2. 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: $[x_0, x_1, ..., x_{12287}]^T$ 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.



the process of L-layer Neural Network nearly to 2-Layer, and the accuray of train set is 98%, while the test set is 80%

thus, the 5-layer neural network has better performance than the 2-layer neural network on the same test set.