Logistic regression lab4

Eng: Samar Shaban

- Problem: Recognizing cat pictures.
- Algorithm: Logistic Regression with neural network mindset
- Performance matrix: Accuracy

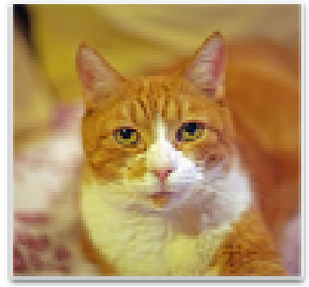


image2vector

$$\begin{pmatrix} 255 \\ 231 \\ \dots \\ 94 \\ 142 \end{pmatrix}$$

/255 → $x_0^{(i)}$

/255 → $x_1^{(i)}$

$\dots$ → $\dots$

/255 → $x_{12286}^{(i)}$

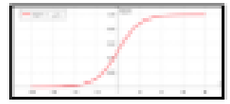
/255 → $x_{12287}^{(i)}$

w_0, w_1, $\dots$, w_{12286}, w_{12287}

$w^T x^{(i)} + b \mid \sigma$ → 0.73

0.73 > 0.5

"it's a cat"



Activate Windov
Go to System in Contr

Mathematical expression of the algorithm:

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

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

The cost is then computed by summing over all training examples:

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

Forward Propagation:

- You get X
- You compute $A = \sigma(w^T X + b) = (a^{(1)}, a^{(2)}, \ldots, a^{(m-1)}, a^{(m)})$
- You calculate the cost function: $J = -\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log(a^{(i)}) + (1 - y^{(i)}) \log(1 - a^{(i)})$

Here are the two formulas you will be using:

$$\frac{\partial J}{\partial w} = \frac{1}{m} X (A - Y)^T$$

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} (a^{(i)} - y^{(i)})$$

- **optimization function:**

The goal is to learn **w** and **b** by minimizing the cost function **J**. For a parameter **θ**, the update rule

$$\theta = \theta - \alpha \ d\theta,$$

where α is the learning rate.

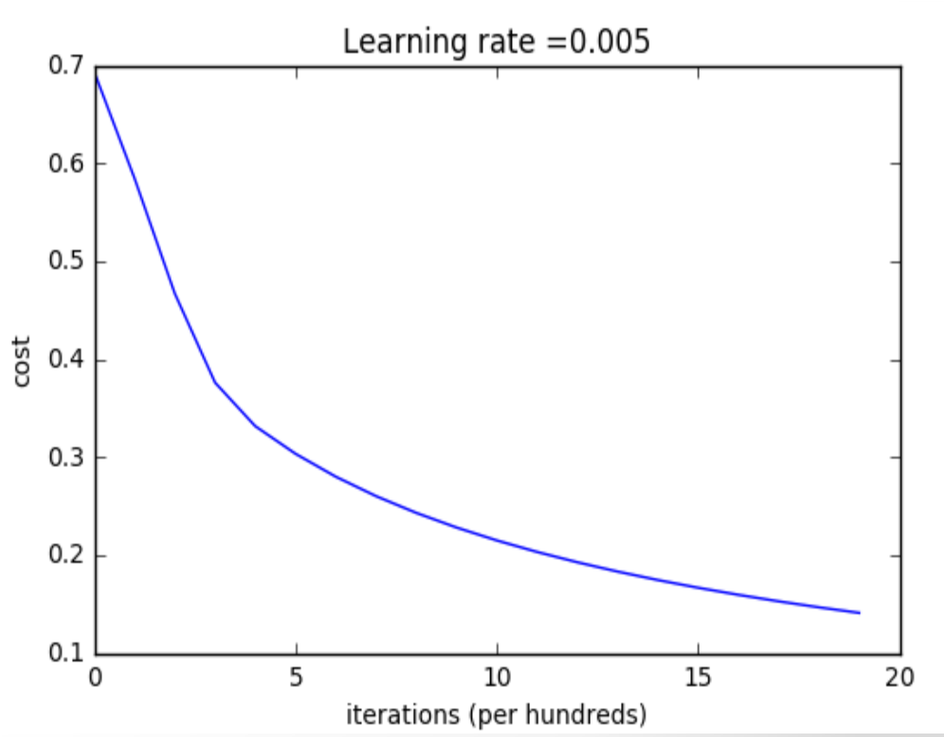
function	parameters	Returned values
load_dataset()		train_set_x_orig, train_set_y_orig, test_set_x_orig, test_set_y_orig, classes
initialize()	dim	w, b
Propagate()	w, b, X, Y	grads = {"dw": dw, "db": db} cost
optimize()	w, b, X, Y, num_iterations, learning_rate, print_cost = False	params = {"w": w, "b": b} grads = {"dw": dw, "db": db} costs
Predict()	w, b, X	Y_prediction
model()	X_train, Y_train, X_test, Y_test, num_iterations = 2000, learning_rate = 0.5, print_cost = False	d = {"costs": costs, "Y_prediction_test": Y_prediction_test, "Y_prediction_train" : Y_prediction_train, "w" : w, "b" : b, "learning_rate" : learning_rate, "num_iterations": num_iterations}

Notes :

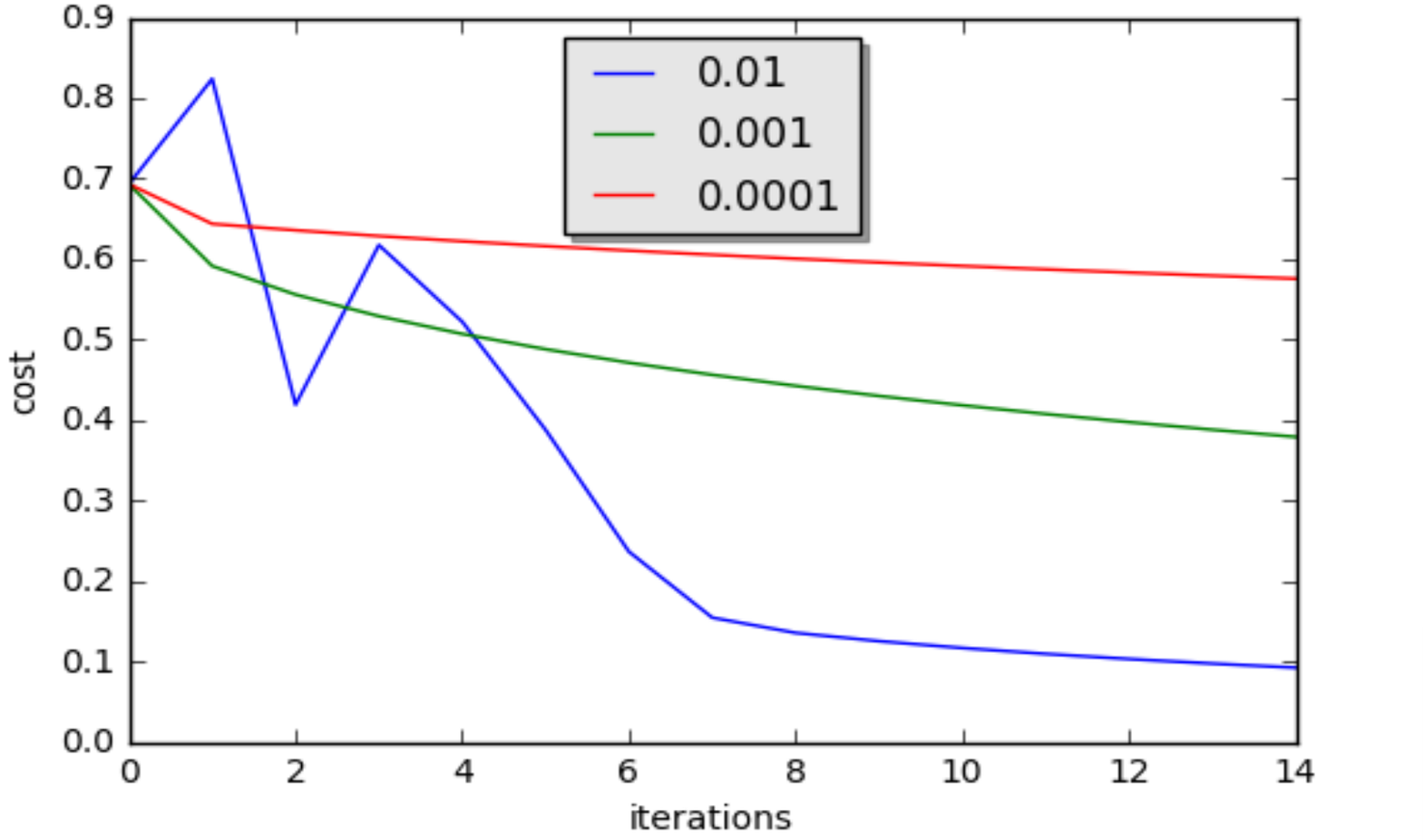
- Training accuracy is close to 100%. The model has high enough capacity to fit the training data. Test accuracy is 70%; It is actually not bad for this simple model. ***Overfitting***

- cost decreasing -->the parameters are being learned.

- The learning rate α determines how rapidly we update the parameters.



- Different learning rates give different costs and thus different predictions results.



- **Implement :**
  - Apply logistic regression as explained on *Titanic Passengers Dataset.*

    Score: 8 of 10

  - Give me some analytical indicators About the Dataset.

    Score 2