G H Raisoni College of Engineering and Management, Wagholi, Pune

Department of AI & DS Engineering

LAB MANUAL

Subject: COMPUTER VISION AND DEEP LEARNING

Class: TY (Computer – 2020 Course)

Examination Scheme:
Practical (INT): 25 Marks
Practical (EXT): 25 Marks
Total: 50 Marks

Prepared BY: PROF. RADHA S SHIRBHATE

Institute Vision & Mission

Vision

Our vision is to achieve excellent standards of quality education by keeping pace with rapidly changing technologies. Our continuous endeavor is to create manpower of global standards with capabilities of accepting new challenges.

Mission

Our efforts are dedicated towards imparting quality and value based education to raise the satisfaction level of our students. Our strength is directed towards creating competent professionals. Our endeavor is to provide all possible and sustainable support to promote research & development activities

Department Vision & Mission

VISION:

To produce global standard ethical professionals, innovators, and entrepreneurs having strong knowledge and urge to learn latest technologies in the field of computer engineering.

MISSION:

The department continuously strives

M1: Pursue excellence in Computer Engineering, able to adapt changing technologies through effective Teaching- Learning Process.

M2: Develop competent professionals for global market with the spirit of self-study, team work, innovation and ethics.

M3: Promote continuous learning, entrepreneurial skills and research.

ASSIGNMENT NO. 1:

TITLE:

Write a program Logistic Regression with Neural Network mindset

PROBLEM STATEMENT:

You are given a dataset (”data.h5”) containing:

- a training set of m_train images labeled as cat (y=1) or non-cat (y=0)
- a test set of m_test images labeled as cat or non-cat
- each image is of shape (num_px, num_px, 3) where 3 is for the 3 channels (RGB).Thus, each image is square (height = num_px) and (width = num_px).

OBJECTIVE:

- Build the general architecture of a learning algorithm, including:
  - Initializing parameters
  - Calculating the cost function and its gradient
  - Using an optimization algorithm (gradient descent)
- Gather all three functions above into a main model function, in the right order.

THEORY:

Welcome to the first (required) programming exercise of the deep learning specializa- tion. In this notebook you will build your first image recognition algorithm. You will build a cat classifier that recognizes cats with 70% accuracy!

Figure 1.2.1 cat

As you keep learning new techniques you will increase it to 80+ % accuracy on cat vs. non-cat datasets. By completing this assignment you will:

- Work with logistic regression in a way that builds intuition relevant to

neural net-works.

- Learn how to minimize the cost function.
- Understand how derivatives of the cost are used to update parameters.

Take your time to complete this assignment and make sure you get the expected outputs when working through the different exercises. In some code blocks, you will find a "#GRADED FUNCTION: functionName" comment. Please do not modify these comments. After you are done, submit your work and check your results. You need to score 70% to pass. Good luck :) !

Packages:

- numpy is the fundamental package for scientific computing with Python.
- h5py is a common package to interact with a dataset that is stored on an H5 file.
- matplotlib is a famous library to plot graphs in Python.
- PIL are used here to test your model with your own picture at the end.

General Architecture of the learning algorithm:

It's time to design a simple algorithm to distinguish cat images from non-cat images. You will build a Logistic Regression, using a Neural Network mindset. The following Figure explains why **Logistic Regression is actually a very simple Neural Net-work!**

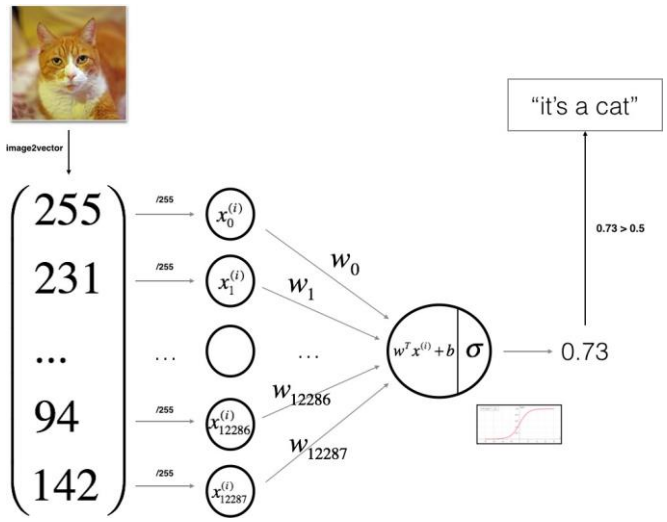

Figure 1.2.2 Principle of Logistic Regression

Mathematical expression of the algorithm:For one example $x^{(i)}$:

$$z^{(i)} = w^T x^{(i)} + b \tag{1.2.1}$$

$$\hat{y}^{(i)} = a^{(i)} = sigmoid(z^{(i)}) \tag{1.2.2}$$

$$L(a^{(i)}, y^{(i)}) = -y^{(i)} \log(a^{(i)}) - (1 - y^{(i)}) \log(1 - a^{(i)}) \tag{1.2.3}$$

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

$$J = \frac{1}{m} \sum_{i=1}^{m} L(a^{(i)}, y^{(i)}) \tag{1.2.4}$$

Key steps: In this exercise, you will carry out the following steps:

- Initialize the parameters of the model
- Learn the parameters for the model by minimizing the cost
- Use the learned parameters to make predictions (on the test set)
- Analyse the results and conclude

Building the parts of our algorithm:

The main steps for building a Neural Network are:

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

You often build 1-3 separately and integrate them into one function we call model().

Optimization:

- You have initialized your parameters.
- You are also able to compute a cost function and its gradient.
- Now, you want to update the parameters using gradient descent.

Merge all functions into a model:

You will now see how the overall model is structured by putting together all thebuilding blocks (functions implemented in the previous parts) together, in the right order.

Exercise: Implement the model function. Use the following notation:

- Y_prediction for your predictions on the test set
- Y_prediction_train for your predictions on the train set

w, costs, grads for the outputs of optimize()

What to remember from this assignment:

1. Preprocessing the dataset is important.
2. You implemented each function separately: initialize(), propagate(), optimize().Then you built a model().
3. Tuning the learning rate (which is an example of a ”hyperparameter”) can make abig difference to the algorithm. You will see more examples of this later in this course!

SAMPLE CODE:

PLATFORM REQUIRED:

Operating System: Windows

Software or Tools: GOOGLE COLAP

CONCLUSION:

Hence we studied the concept of Logistic Regression with Neural Network mindset along with accuracy is 70% for the dataset CatNoncat.