INTRODUCTION

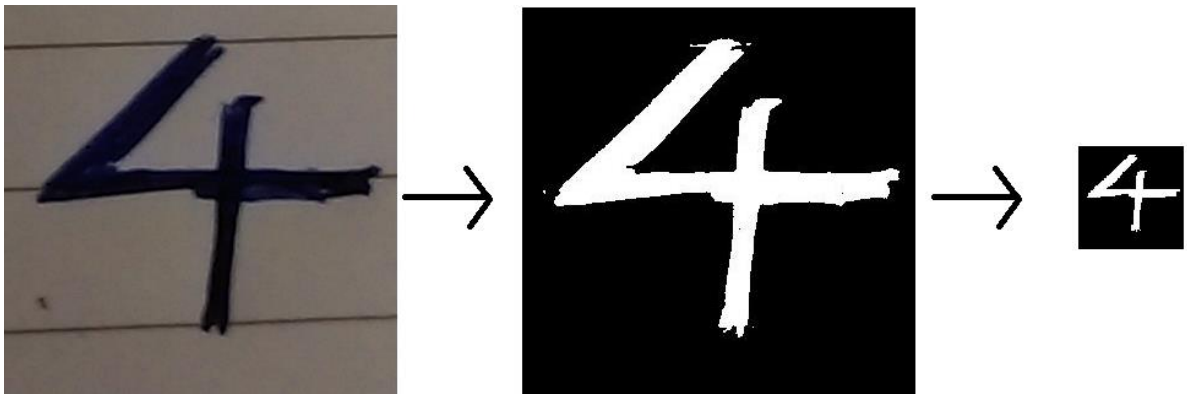
Optical character recognition (optical character reader) (OCR) is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text. It is widely used as a form of data entry from printed paper data records, whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation. It is a common method of digitizing printed texts so that it can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as machine translation, text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

This project envisages the use of logistic regression to classify characters. All the testing and simulations are done using Matlab.

ACQUIRING THE TRAINING DATA

The logistic regression algorithm requires the training data to be stored in a variable. Each character requires numerous versions of images so that the learning algorithm can efficiently classify the characters. In our project we have limited the characters to, ‘1’,’2’,’3’,’4’ and ‘5’. For convenience we have taken 90 images of each character in different orientations and sizes to comprise of our training data. The following steps were carried out to obtain our training data:

1. A picture of the handwritten character was taken.
2. Using colour thresholding, a binary image of the character was obtained comprising of only 1 and 0 values.
3. The image was then resized to a 50x50 pixel image and the character in the image was manipulated to form 90 images of different orientation and sizes.
4. The images were then sequentially taken into Matlab flattened out into a 2500x1 matrix from a 50x50 matrix and then stored into a storage variable. Each image was stored in each row of the storage matrix of dimension 2500x450.
5. Another matrix of dimension 450x1 was created to store the value of the character in the image.

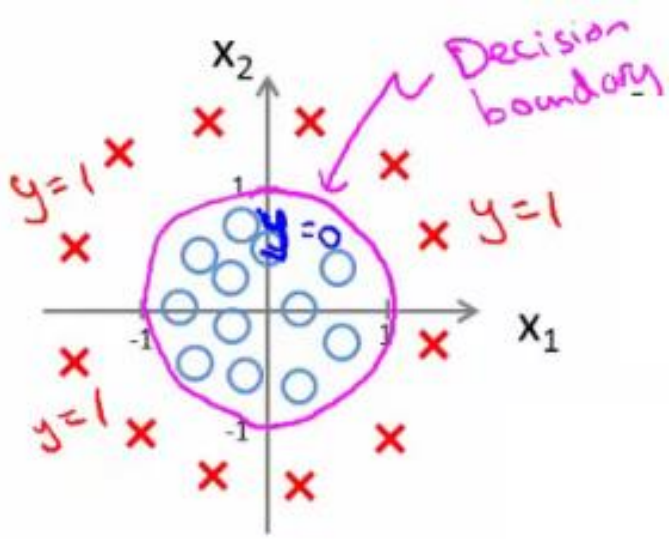


APPLYING LOGISTIC REGRESSION

Logistic regression is a machine learning algorithm used for classification problems. In our project it carries out the task of classifying new input into the various characters based on the training data. For any new input, if it is trained correctly, the algorithm will be able to classify the handwritten character at a high success rate.

To understand logistic regression let us take an example.
Say we have two features x1 and x2, and the values of these parameters are plotted as shown in the figure. These parameters correspond to two values, when y=0 and y=1. The logistic regression algorithm draws a decision boundary so that our new input, if it lies within the boundary or outside it, we can know the state of the input (if y=0 or y=1).

This concept has been used to classify the images with 2500 (the image is compressed into a 2500x1 matrix) features, when y represents the value of the respective characters.



The logistic regression hypothesis is defined as:

$$h_\theta(x) = g(\theta^T x)$$

where function g is the sigmoid function. The sigmoid function is defined as:

$$g(z) = \frac{1}{1 + e^{-z}}.$$

The matrix theta represents the parameter values that form the decision boundary.
The matrix x represents the input matrix.

We find the theta matrix values by minimizing the cost function. The cost function represents the deviation of a point in the data from the best decision boundary. More the deviation of a data point from this decision boundary, greater is the cost. By minimizing the function, we are left with the optimum values of theta to construct the decision boundary.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left[-y^{(i)} \log(h_\theta(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right],$$

and

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

After computing the above and repeating the process for a number of iterations, we are left with a matrix theta containing the parameters that fit the decision boundary.

IMPLEMENTATION

After the system is trained, a new input can be properly classified by the algorithm, using the hypothesis shown in the previous section. The new input must undergo all the processes that the training data used so that computation can be simplified to a great extent.

In the end, we are left with a matrix containing the probabilities of the input to be classified into each character. The index with largest probability is taken into consideration and outputted as that character. When the input is an image containing the character '4', the output is shown in the figure below.

```
>> implementation

p =

    0.0000
    0.0001
    0.0786
    0.9527
    0.0352

The number is most likey:4>>
```

CONCLUSION

Concepts learnt doing this project:

1. Image manipulation
2. Image thresholding
3. Logistic regression
4. Various Matlab functions
5. Manipulation of data in Matlab