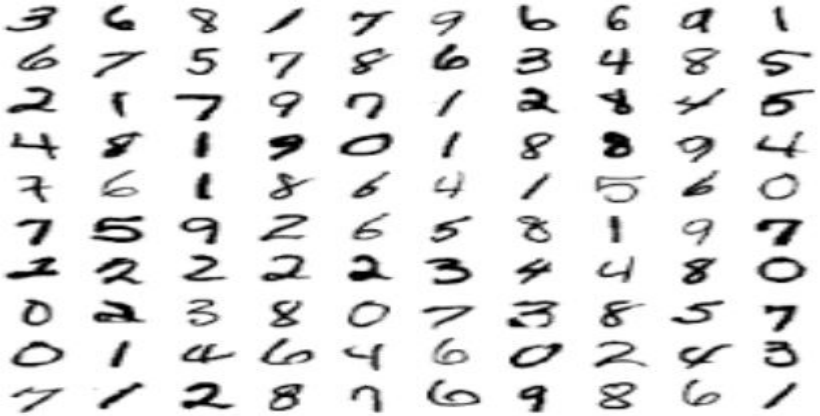
A Project Report On :

HANDWRITTEN DIGIT ReCOGNITION



Submitted by :

Mithun Kumar

Branch: Computer Science And Technology

College : Indian Institute of Engineering Science and Technology

INTRODUCTION:

In this paper we present an program or way to tackle the recognition of human handwritten digits. The set of codes proposed here consists of data set from the library used . Combined with clustering techniques, we can build artificial intelligence system which can automatically segment individual digit from images and find its corresponding label.

Automatic recognition of human handwritten digits was a mysterious job to me. Different people have very different writing style, even digits of a same person written in different time are not identical. How does artificial intelligence deal with the infinite possibility of different shapes of digits, given only an image? Since now I have taken the Machine Learning course and acquired knowledge in this field, I am able to tackle this problem with a matlab program.

The program I implement will mainly focus on identifying 0-9 from segmented pictures of handwritten digits. The input of my program is a gray level image, the intensity level of which varies from 0 to 255. For simplicity, input images are pre-treated to be of certain fixed size, and each input image should contain only one unknown digit in the middle. These requirements are not too harsh because they can be achieved using simple image processing or computer vision techniques. In addition, such pre-treated image data set are easy to obtain.

The output of my program will be the corresponding 0-9 digit contained in input image. the method used is Nearest Neighbor Classifier that stores the whole training set and classify new input case by case. The input layer contains the same number of units as the number of pixels in input image.

We will use the sklearn package to save the classifier in a file so that we can use the classifier again without performing training each time. Calculating HOG features for 70000 images is a costly operation, so we will save the classifier in a file and load it whenever we want to use it. As discussed above sklearn.datasets package will be used to download the MNIST database for handwritten digits . We will store our HOG features and labels in numpy arrays. The next step is to download the dataset using the sklearn library .

Sklearn library is used to download the dataset . load_digits dataset is used.

Using the command : from sklearn.Datasets import load_digits

Assumption during testing

There are a few assumptions, we have assumed in the testing images –

1.The digits should be sufficiently apart from each other. Otherwise if the digits are too close, they will interfere in the square region around each digit. In this case, we will need to create a new square image and then we need to copy the contour in that square image.

2. For the images which we used in testing, fixed thresholding worked pretty well. In most real world images, fixed thresholding does not produce good results. In this case, we need to use adaptive thresholding.

Machine learning is about learning some properties of a data set and then testing those properties against another data set. A common practice in machine learning is to evaluate an algorithm by splitting a data set into two. We call one of those sets the training set, on which we learn some properties; we call the other set the testing set, on which we test the learned properties.

The dataset which we taken consist feature and label.
Feature is every single pixel value of digit image and label is the target .

scikit-learn comes with a few standard datasets, for instance the iris and digits datasets for classification and the boston house prices dataset for regression.

In the following, we start a Python interpreter from our shell and then load the iris and digits datasets. Our notational convention is that $ denotes the shell prompt while >>> denotes the Python interpreter prompt:

```
$ python
 >>> from sklearn import datasets
 >>> digits = datasets.load_digits()
```

A dataset is a dictionary-like object that holds all the data and some metadata about the data. This data is stored in the .data member, which is a n_samples, n_features array. In the case of supervised problem, one or more response variables are stored in the .target member. More details on the different datasets can be found in the dedicated section. For instance, in the case of the digits dataset, digits.data gives access to the features that can be used to classify the digits samples:

```
>>> print(digits.data)

[[ 0.   0.   5.   ...  0.   0.   0.]
 [ 0.   0.   0.   ... 10.   0.   0.]
 [ 0.   0.   0.   ... 16.   9.   0.]
   ...
 [ 0.   0.   1.   ...  6.   0.   0.]
 [ 0.   0.   2.   ... 12.   0.   0.]
 [ 0.   0.  10.   ... 12.   1.   0.]]
```

and digits.target gives the ground truth for the digit dataset, that is the number corresponding to each digit image that we are trying to learn:

```
>>> digits.target
array([0, 1, 2, ..., 8, 9, 8])
```

Shape of the data arrays

The data is always a 2D array, shape (n_samples, n_features), although the original data may have had a different shape. In the case of the digits, each original sample is an image of shape (8, 8) and can be accessed using:

```
In [3]: digits.images[0]

Out[3]: array([[ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.],
               [ 0.,  0., 13., 15., 10., 15.,  5.,  0.],
               [ 0.,  3., 15.,  2.,  0., 11.,  8.,  0.],
               [ 0.,  4., 12.,  0.,  0.,  8.,  8.,  0.],
               [ 0.,  5.,  8.,  0.,  0.,  9.,  8.,  0.],
               [ 0.,  4., 11.,  0.,  1., 12.,  7.,  0.],
               [ 0.,  2., 14.,  5., 10., 12.,  0.,  0.],
               [ 0.,  0.,  6., 13., 10.,  0.,  0.,  0.]])
```

The simple example on this dataset illustrates how starting from the original problem one can shape the data for consumption in scikitlearn

Loading from external datasets

To load from an external dataset, please refer to loading external datasets.

In the case of the digits dataset, the task is to predict, given an image, which digit it represents. We are given samples of each of the 10 possible classes (the digits zero through nine) on which we fit an estimator to be able to predict the classes to which unseen samples belong.

In scikit-learn, an estimator for classification is a Python object that implements the methods fit(X, y) and predict(T).

scikit-learn comes with a few standard datasets, for instance the iris and digits **datasets** for classification and the boston house prices **dataset** for regression. A **dataset** is a dictionary-like object that holds all the data and some metadata about the data.

Conclusion:

Handwritten Digit Recognition can be coded in many ways . a simple way is to use sklearn method . here datasets are imported from the library itself . Scikit-learn is probably the most useful library for machine learning in Python. It is on NumPy, SciPy and matplotlib, this library contains a lot of effiecient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.Please note that scikit-learn is used to build models. It should not be used for reading the data, manipulating and summarizing it. There are better libraries for that (e.g. NumPy, Pandas etc.)

By doing this project, I have practiced using what I have learned in the Machine Learning course. My program probably does not beat the state of the art in handwritten digit recognition. However, I have for the first time observed the practical problems of using sklearn . I have learned about its various components . I have dealed many practical problems associated with sklearn .

REFRENCES:

https://www.wikipedia.org/

https://scikit-learn.org/stable/

https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/

https://scipy-lectures.org/packages/scikit-learn/index.html