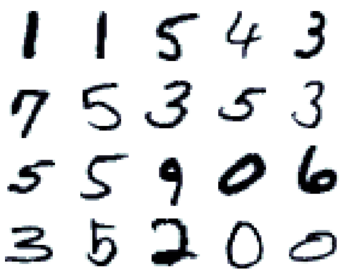
Recognizing Hand-Written Digits

January 2018



This Lab aims at developing a program able to recognize the hand-written digits of the MNIST dataset. [1]

1 Getting familiar with the data

The dataset can be downloaded from the lecture website. The following piece of code allows to load the data:

```python
import pickle
import gzip
import numpy

# Load the dataset
with gzip.open('mnist.pkl.gz', 'rb') as ifile:
    train, valid, test = pickle.load(ifile, encoding="latin-1")
```

The corpus is split into a training set, a validation set and a test set. Each dataset is made of an array gathering to objects:

- an array of images, each image is an array of 784 elements (a value between 0 and 1 corresponding to the gray level of a pixel);
- an array of integer corresponding to the label (a digit between 0 and 9). The i-th element of the array is the label of the i-th image of the previous array.

[1] `http://yann.lecun.com/exdb/mnist/`

The following code can be used to visualize an image:

```python
import matplotlib.pyplot as plt

im = train_set[0][0].reshape(28,28)

plt.imshow(im, plt.cm.gray)
plt.show()
```

1. What is the size of the images? How are they represented?

2. The first step consists in binarizing the images using a threshold ϵ. The simplest thresholding method, we will consider in this work, replace each pixel in an image with a black pixel if the image intensity $I_{i,j}$ is less than some fixed constant T, or a white pixel if the image intensity is greater than that constant. The following function can be used to binarize an image:

```python
def binarize(im, thres=0.5):
    """Return a binarized copy of an image using a given threshold
    level.

    See the section ‘Boolean or \mask" index arrays’ in
    http://docs.scipy.org/doc/numpy/user/basics.indexing.html
    """
    # Create a new zeros array with adequate shape
    res = np.zeros(im.shape)
    # Find values above threshold in im an set them to 1 in `res`
    res[im >= thres] = 1
    return res
```

   Explain how it works. In the following we will consider that $T = .5$[2]

3. The horizontal histogram of a binary image is an array that has as many elements as the number of rows of the image. The i-th value of this array, is the number of “active” pixels in the i-th row of the image. Write a function that computes the horizontal histogram of an image. **Hint:** look at the `numpy.sum` function.

4. Look at the horizontal histogram of the picture of a 1 and of 5. Do you think that an histogram will help us discriminating between these two labels?

5. By analogy: write a function that computes the vertical histogram of an image.

[2]Considering $T = .4$ will lead to interesting results.

6. Write a function that takes as input a list of images and compute the mean image of this list: the value of the pixel at position (x, y) is the mean the n pixel values at position (x, y) of the input images.

2 Classification

We aim at comparing the prediction performance of two classification methods considering different feature sets. In the following we will always consider binary images and use the Euclidean norm as a distance.

7. Represent the error rate achieved by a k-nn classifier for $k \in [\![1, 10]\!]$ on the three datasets

8. How to choose the value of k to achieve the best performance?

9. Implement a minimum distance classifier. Which error rate does it achieve?

10. Compute the confusion matrix of the k-nn classifiers. The confusion matrix is $n \times n$ matrix where n is the number of classes. The (i, j) element of this matrice is the number of times the i-th class has been predicted instead of the j-th class. What does the confusion matrix show?

Considering only the optimal value of k, what are the prediction performance when

11. the raw images are considered (ie. non-binarized images);

12. the vertical and horizontal histogram are considered as features.