Digits Lab - Week 5

There are three files for this lab.

1. This pdf : `Assignment1.pdf`
2. The workbook : classifying hand-written digits - `digits.ipynb`.
3. The images : `digits.zip`.

All files are available on Moodle. Complete all work in the corresponding notebook, typing any answers to questions in a Markdown cell type.

When you are asked to answer a question, a one-line answer is not enough, you need to give an explanation. Put answers in markdown cells. The interpretation of information and results is at least as important as the building of models.

Since this is image classification to a degree, a Convolutional Neural Network is what you'd probably want to use in reality, but I want to see if the techniques we know so far could be effective.

Hand Written Digits

You need to get Part 1 preparing the data working correctly (the reading in of the data), however for any other parts/subparts if you have difficulty skip to the next part.

You will be using a set of 5,000 images that have been classified into digits $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. There are 10 classes and 500 samples per class. You will use this model to select a classifier by comparing k-Nearest Neighbours and Logistic Regression for this data set. Each image is 20 pixels by 20 pixels.

1 Preparing the Data

- The `digits.ipynb` workbook has the initial imports.
  - `glob.glob('data/*/*.txt')` will give you a list of all files matching within the brackets
  - `Image.open(filename)` will open an image file. We can extract each pixel's information by putting this into an array. `np.array(Image.open(filename)` will be a 20x20 array.

1. First step is to make the matrix X (all the attributes) and the vector y (the corresponding class).
   - Create a filelist (using `glob`
   - Create an array of all the images from this file list. (Try to do this in one line, avoiding loops, but use them if you have to so you can proceed through the rest of the lab, or ask your lecturer for help)
   - Create the array of classes, ensure this is ordered the same way as X. Do this whatever way you wish. Remember, you cannot guarantee filelist is ordered or there is an equal distribution of categories.
     - You can get the response variable from the filename, use slices

```
a = [s.split() for s in ' '.join(filelist).split('\\') if s]
```

May Help

- Another alternative method is, sort the filelist before creating the X array, then the first so many will be 0, the next lot will be 1, etc. However, you will need to ensure you have the right number of category 0, 1, 2 etc manually!

- If you type y in its own cell it should look something like this `array([0, 0, 0, ..., 9, 9, 9])`

2. Are X and y the right shapes? X should be a 2D array, y should be a 1D array. Reshape as needed.

3. If everything is set up correctly, the code commented as `##Printing Examples` will output a random selection of 10 examples for each class.

4. What type of colours are the pixels in the image? What range of numbers do they take?

5. When dealing with image data, often it is better to have the numbers in the range 0-1 rather than 0-16 or 0-255 so you should alter the X array by dividing by the right number.

6. You will use the same training set, validation set and test set for all models. Use train_test_split method with `random_state` set so that you get the same split everytime you run the workbook.

2 k-Nearest Neighbours

Let's start with k-Nearest Neighbours as that's the last one we talked about in lectures (unless I've covered SVM by now!)

1. Create a k nearest neighbours model using $k = 5$. Properly train and score the model on Training and validation sets.

   - If it is a good score.
   - Also can you think of some reasons the NB model would not be a good classifier? What assumptions are made and are they good assumptions?

2. Using `model.predict` count how many incorrect predictions on the test set were made by the model.

3. Print a classification report

   - Which classes have good/bad precision/recall?

4. Print the confusion matrix. Comment

   - Which class has been misidentified the most - false negatives? (e.g. images that should have been in class 8 but got put in class 0,1,2,3,4,5,6,7 or 9)
   - Which class has the most false positives? (e.g. an image was identified as class 8 but this was not correct)
   - Do this using code.

5. Implement a kFold Cross Validation to find the best k. (use 5 folds).
    - After finding the *best* k, train a model using the training set using that k
    - What is the final accuracy score (for the validation set) with the k you have now chosen? (save this score)
    - Print a classification report and comment on it.

3 Logistic Regression

1. Repeat the procedure using Logistic Regression as your model. Using kFold Cross Validation to choose your amount of regularisation.
2. Print the confusion matrix. Comment.
3. Print a classification report and comment on it.
4. Keep the accuracy score for the validation set on the *best* Logistic Regression model.

4 Finalising

1. What was the model with the best validation score?
    - Don't just say k-nearest neighbours (or Logistic Regression) as this is not the model, just the model type.
    - You need to say for example "The model with the best validation score was the model of the form k-nearest neighbours trained with $k = 6$ neighbours."
2. Now make your final model. Combine the training and validation set and build a new model of the form you want.
3. Finally, you can get the test-score from that model and check its classification report and confusion matrix.