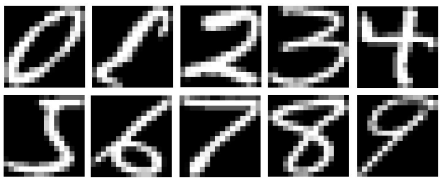
School of Computer Science | The University of Manchester

Handwritten Digit Recognition
Lab2

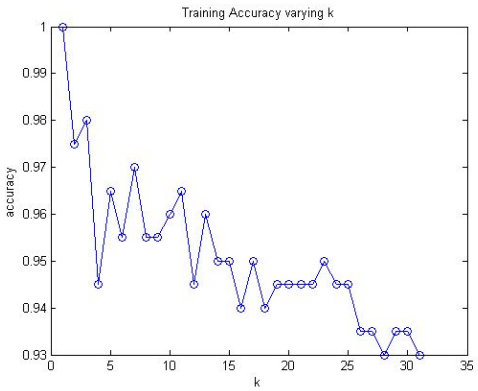
Leonardo Claudio de Paula e Silva
9844904

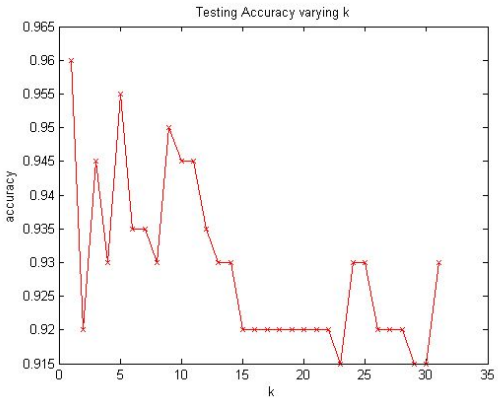
Part 0

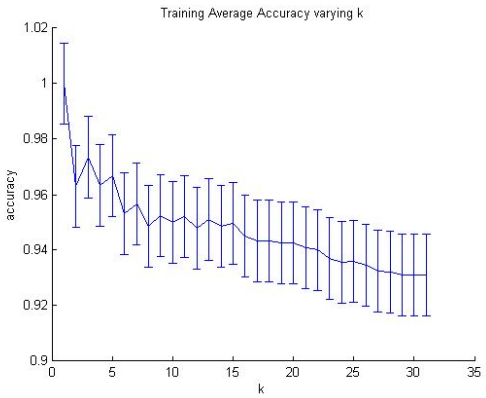


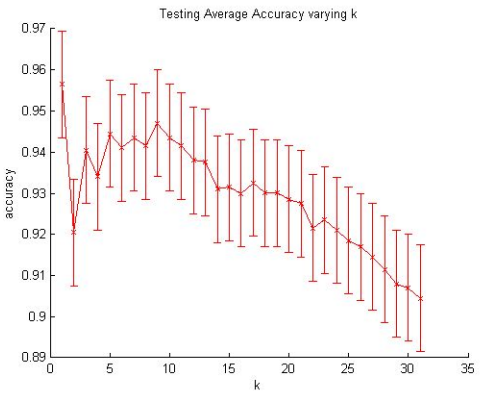
Part 1

Basically, the algorithm compare the class of the k nearest neighbors for the element and classify the element as the class the most appeared. As k gets bigger the number os neighbors to compare increases the consistency increases since it reduces the effect of noises on the classification, however the boundaries between classes becomes less distinct. When k is equal to one it is called the nearest neighbor algorithm and it probably leads to a over-fitting.
To define the value of k that best fits the problem is not straightforward, actually, as the data is generated randomly the best value for k may not be the same at all interaction, hence it is needed an average accuracy to obtain the best value.



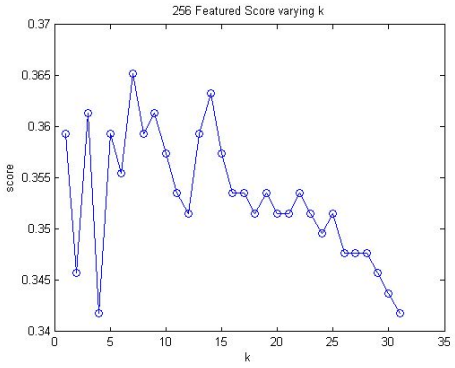


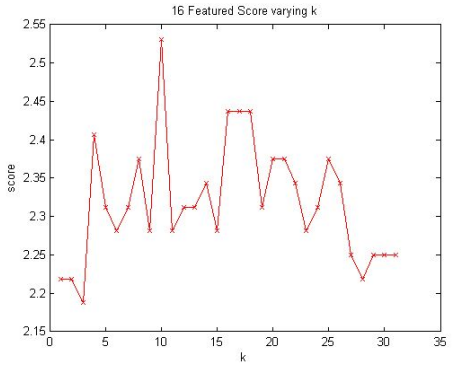




Part 2

Score is a methodology to compare the amount of information (memory) necessary to get a accuracy. If the score is higher it means that less memory is necessary to obtain a certain accuracy which is the goal. As it seems, the accuracy goes down to 0.30 with 16 features counting the number of gaps in each row. Therefore, if I use less features my accuracy is lower but the amount of data to process and to store is much lower too.





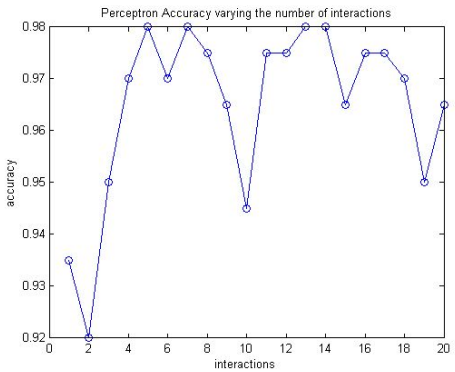
Analysing the whole picture for the 256 features, there is 27, 24 and 11 true positives for '3','6' and '8' digits respectively. Therefore the '8' digit has the greatest error. For the 16 features data we have 24, 24 and 16. Once again, the '8' digit has the greatest error.

$$CM256 = \begin{bmatrix} 27 & 19 & 19 \\ 30 & 24 & 17 \\ 24 & 29 & 11 \end{bmatrix}$$

$$CM16 = \begin{bmatrix} 24 & 24 & 18 \\ 27 & 24 & 17 \\ 30 & 20 & 16 \end{bmatrix}$$

Part 3

The KNN algorithm is a instant training algorithm, in that case there is no training there is just a comparison between the element and the k nearest neighbors. Despite of this great advantage of a instant result the amount of memory that is needed is huge. It is necessary to store the whole dataset in memory in order to use this method. When using the perceptrons, it is necessary to really train the machine, so there is a time cost, however it is not necessary to store the dataset once the machine is trained, hence there is gain in memory.



For full report go to https://copy.com/hIsm3GcRNxybZDJJ