CW2 CST-3170

Ronit Rai	M00872394
CST 3170	**Chris Huyck**
UCI Digit Categorization	16/02/23

For this task I have used 2 Different Algorithms, this allows me to compare the different approaches and differences in the success percentage as well as the quality of results.

Euclidean Distance

Firstly Euclidean Distance is used, simply by comparing each data item in the test set to all of the data items in the training set, the comparison is done using the simple Euclidean distance formula where the 64 attributes are used as dimensions. The total number of correct predictions are returned and a percentage is calculated. With this solution the best results are obtained, an average success percentage of 98.26% is returned. Although a near-perfect percentage is returned, this algorithm is very computationally intensive with a O(n^3) Big O Notation, and would not be that useful with real world data.

Using Data Set 1 as Test Set

- 2810 Data Items
- 98.47% Success Rate
- O(n^3)

Using Data Set 2 as Test Set

- 2810 Data Items
- 98.04% Success Rate
- O(n^3)

SOM (Self Organizing Maps)

The Second Algorithm used is Self-Organizing Maps. This is a un-supervised learning algorithm. Firstly variables such as the learning rate & maximum iterations are assigned using which the weights are calculated. One Data Set is passed which is used to calculate the weights, a random index is chosen from where its randomly assigned weight is compared with all other data items in the array.

The distance, neighborhood radius & influence are 3 parameters used to update the weights. The Distance is the simple Euclidean Distance b/w the current data item and the comparison data item, The neighborhood radius is the parameter that controls the extent to which the weights of the winning neuron (closest matching data item) are updated, the Influence is in charge to determining the effect on neighboring data items' weights. These values are calculated using a Gaussian Function.

$$f(influence) = e^{-distance*distance/2*radius*radius}$$

$$f(neighbourhoodRadius) = rows/2 * e^{-iter/(maxIter/2)}$$

This particular algorithm gives me a very varied set of results since there are many variables to play around with, the best results I have gotten have been b/w 85% and 90% depending on the Learning Rate, Number of Iterations & Number of Dimensions used for the Weights.

The Big O Notation for this Algorithm is usually O(n*2), much more efficient compared to the Euclidean Distance Solution.

Testing Results

Fixed Values

1. Data Items → 2810
2. Data Attributes → 64
3. Weight Count per Item → 1

Variables

1. Learning Rate
2. Number of Iterations

Best results are obtained on average when Learning Rate is 1.0 & the number of iterations are increased, 88+% on average using 15,000 iterations. Best Result of 91.03% (91.03202846975088) is obtained when Learning Rate is 0.02 and Number of Iterations is 15000. Similar results are obtained when switching up the test and training data. Average Success Rate is about 90%.

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/91b84818-bcd6-4cc7-a0c7-f00c4e823ab1/reresults.txt