Including Metric Space Topology in Neural Networks Training by Ordering Patterns

Cezary Dendek[1] and Jacek Mańdziuk[2]

[1] Warsaw University of Technology, Faculty of Mathematics and Information Science, Plac Politechniki 1, 00-661 Warsaw, POLAND,
dendekc@student.mini.pw.edu.pl,
[2] Warsaw University of Technology, Faculty of Mathematics and Information Science, Plac Politechniki 1, 00-661 Warsaw, POLAND,
(phone: (48 22) 621 93 12; fax: (48 22) 625 74 60)
mandziuk@mini.pw.edu.pl,
WWW home page: http://mini.pw.edu.pl/~mandziuk/

Abstract. In this paper a new approach to the problem of ordering data in neural network training is presented. According to conducted research, generalization error visibly depends on the order of the training examples. Construction of an order gives some possibility to incorporate knowledge about structure of input and output space into the training process. Simulation results conducted for the isolated handwritten digit recognition problem confirmed the above claims.

1 Introduction

The problem of optimal ordering of the training data has a great meaning in sequential supervised learning. It has been shown ([1],[2]), that improper order of elements in the training process can lead to catastrophic interference. This mechanism can also occur during each training epoch and disturb neural network training process. Random order of elements prevents from interference but can lead to non-optimal generalization. Consequently, for example, most of efficient algorithms for training RBF networks arbitrarily choose initial patterns ([3]).

In this paper a new approach to patterns ordering is proposed and experimentally evaluated in the context of supervised training with feed-forward neural networks. The idea relies on *interleaving two training sequences: one of particular order and the other one chosen at random.*

In order to show the feasibility of this approach four models of an order are defined in the next section together with a sample test problem - isolated handwritten digit recognition. Numerical results of proposed *interleaved* training are presented in Sect. 2. Conclusions and directions for future research are placed in the last section.

Input and output spaces of a network can be considered as metric spaces. It is always possible to introduce a metrics since each of them can be immersed in $\mathbb{R}^n$ (where n is a space dimension) with natural metrics

$$M : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+ \cup \{0\}, \qquad M : ((x)_{k=1}^n, (y)_{k=1}^n) \mapsto \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}.$$

Moreover, if some other knowledge about the data is possessed - e.g. if input data consists of $p, (p > 1)$ values of different scales - metrics with normalization or non-euclidean metrics may be used, which would model the space considerably better. In such a case it is possible to divide a space into p subspaces and calculate the following metrics:

$$M : (x, y) \mapsto \sqrt{\sum_{i=1}^{p} \Big(\frac{d_i(x, y)}{\overline{d_i}} \Big)^2}, \tag{1}$$

where $d_i(x, y)$ denotes the distance between elements x and y according to the i-th metric and $\overline{d_i}$ represents the average pairwise distance between elements belonging to the i-th scale data.

After choosing and normalizing metrics on input and output spaces it is possible to introduce metrics on pattern space as it was done on space divided into subspaces.

1.1 Four models of an order

In this section four schemes of ordering training patterns together with their characteristics are introduced.

Let input and output spaces be denoted by I and O, resp., and let $\{T_k\}$ be the set of training patterns. The models presented below rely on the fact that given a metrical space of patterns it is possible to determine a pattern that is the nearest to the center of the average probability of occurrence - analogously to the *mass center point*.

Model I. Let us denote by S_k^I a sum of distances from a given element T_k to the rest of elements of the set:

$$S_k^I = \sum_{l=1}^{n} M(T_k, T_l)$$

A sequence of q training patterns $(T_l)_{l=1}^{q}$ that fulfils the following set of inequalities:

$$\forall_{1 \le l \le q-1} \qquad S_l^I \ge S_{l+1}^I \tag{2}$$

is called *ordered set of model I*. A sequence created with rule (2) will begin with outlying patterns and end with close-to-average ones. Since the ending of the sequence finally tunes weights of the network (and if not randomly chosen can have a biased impact on the final network's weights) it is important to characterize these average elements. In the space of patterns an ending of the sequence is expected to concentrate on the following two kinds of neighborhoods:

1. *Global maxima of probability density*. In such a neighborhood an average distance should be minimized.
2. *Geometrical centers*. These points minimize the sum of distances to all other points. If probability of patterns is uniformly distributed the sequence ending would be concentrated on geometrical centers.

In case of multicluster data it is expected that the training sequence ending would be dominated by elements of one of the clusters (except for the cases of symmetrical distributions of clusters). In such a case the sequence ordered in the above way will generalize an approximated function better than a randomly ordered sequence only on elements of preferred cluster.

Since the construction of an ordered set according to *model I* is straightforward its description is omitted.

Model II. Given a metrics M defined on pattern space and a set $\{T_k\}$ an average pairwise distance S_n^{II} between the first n elements of the sequence can be expressed as:

$$S_n^{II} = \frac{2}{(n-1)n} \sum_{k=1}^{n} \sum_{l=k+1}^{n} M(T_k, T_l).$$

A sequence of q training patterns $(T_l)_{l=1}^{q}$ that fulfils the set of inequalities:

$$\forall_{1 \leq l \leq q-1} \qquad S_l^{II} \geq S_{l+1}^{II} \tag{3}$$

is called *ordered set of model II*. Similarly to the previous model a sequence created with rule (3) is expected to prefer outlying patterns at the beginning of the sequence and place the average ones at the sequence ending. Rule (3) is more sensitive to geometrical centers than probability centers compared to rule (2). A reason for such statement is an observation that elements in the sequence ordered using rule (3) that occur after given element do not have an influence on its position (as if they had been removed from the set). What is more, a selection of an element according to presented algorithm implies that the difference in the average distance after selection is minimal - the change of geometrical center of a set should also be small. Removal of an element changes local density of probability.

Algorithm for ordering a set in Model II. Given set $\{T_k\}$ can be ordered to sufficiently approximate *ordered set of model II* with the use of the following algorithm:

1. Put all q elements in any sequence $(T_l)_{l=1}^{q}$.
2. Create an empty sequence O.
3. Create distance array $D[1..q]$:

$$\forall_{1 \leq l \leq q} \qquad D_l := \sum_{k=1}^{q} M(T_l, T_k)$$

4. Choose a minimal value of element of D:

$$v := min_{1 \leq l \leq q} \qquad D_l.$$

5. Pick one element k from the set $\{1 \leq l \leq q \,|\, D_l = v\}$.
6. Update distance matrix:

$$\forall_{1 \leq l \leq q} \qquad D_l := D_l - M(T_k, T_l)$$

7. Take element T_k out of sequence T and place it at the beginning of sequence O.
8. Remove element D_k from distance array.
9. Put $q := q - 1$.
10. Repeat steps 4-10 until $q = 0$.

Model III. *Ordered set of model III* is obtained by reverting *ordered set of model I.*
Model IV. *Ordered set of model IV* is obtained by reverting *ordered set of model II.*

1.2 Test problem

In order to test an influence of training data ordering on the learning process, a sample problem consisting in isolated handwritten digits recognition was chosen. The pattern set consisted of 6000 elements, randomly divided into training set T, $|T| = 5500$ and test set V, $|V| = 500$. Binary $\{0, 1\}$ input vectors of size 19×17 represented bitmaps of patterns, and the 10-element binary $\{0, 1\}$ output vector represented the classification of the input digit (i.e. exactly one of its elements was equal to 1). All patterns were centered. It should be noted that no other preprocessing took place. In particular digits were not scaled, rotated or skewed appropriately. A detailed description of this data set and results achieved by other recognition approaches can be found in [4].

An ensemble of neural networks with 1 hidden layer composed of 30 neurons was trained using backpropagation method. Both hidden and output neurons were sigmoidal.

Input subspace became metrical with the use of the following metrics:

$$I(v, w) = min_{x,y \in \{-2,-1,0,1,2\}} H(v, R(x, y, w)) + |x| + |y|$$

where $H(\cdot, \cdot)$ denotes Hamming distance, and $R(x, y, w)$ denotes translation of vector w by x rows and y columns. In the output subspace a discrete metrics $O(v, w)$ was used. Based on metrics defined on subspaces a metrics on pattern space was defined according to (1) as follows:

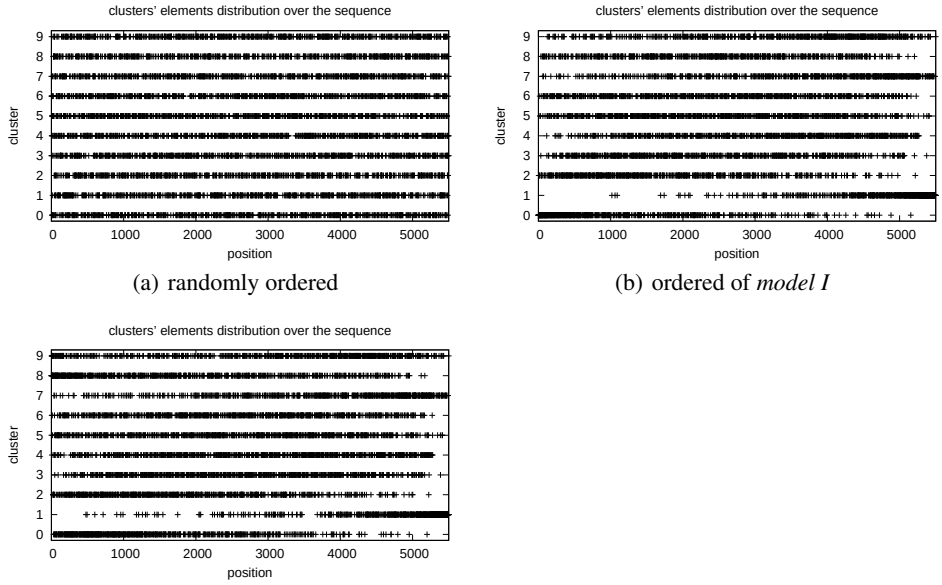
$$M : (x, y) \mapsto \sqrt{\Big(\frac{I(x, y)}{\overline{I}}\Big)^2 + \Big(\frac{O(x, y)}{\overline{O}}\Big)^2}.$$

For the training set it was obtained $\overline{I} = 62.55$, $\overline{O} = 0.9$.

2 Results

All numerical results concerning RMSE and STDEV are presented as the averages over 100 networks, each with randomly selected initial weights. Unless otherwise stated each training period was composed of 600 epochs. For comparison purposes all figures representing one pass of training/testing for different orders of the training data are presented for the same, randomly selected network (i.e. with the same initial choice of weights).

According to previously formulated hypothesis in case of ordered sequences elements of particular clusters were not uniformly distributed over the sequence, which is



(a) randomly ordered

(b) ordered of *model I*

(c) ordered of *model II*

Fig. 1. Clusters' elements distribution over sequences

illustrated in Fig. 1. For example, elements representing digits 1 and 7 are concentrated at the endings of both ordered sequences (Fig. 1(b) and Fig. 1(c)) and elements representing 0 and 2 are located mainly at the beginnings, whereas distributions of all digits in case of random order (Fig. 1(a)) are uniform. Distributions of ordered sequences are similar to each other, but they remarkably differ on digits 8 and 9.

2.1 Initial results for pure random and ordered training data

The case of randomly ordered training data (henceforth referred to as *pure random* case) proves that the considered problem can be solved using assumed network architecture and learning algorithm. This case also provides a possibility of comparison between ordered training models and the pure random one. The plot of RMSE of the training and test data in pure random order case are presented in Fig. 2.

The plots of RMSE of the network trained with sequence ordered according to *model II* are presented in Fig. 3. **It can be concluded from the figure that convergence of training is worse compared to random order**.

In hope to improve the convergence of the training process switching of training sequences with a random one was tried. Fig. 4 presents changes of RMSE in case the first 300 training epochs was performed with the sequence defined according to *model IV*, which was then replaced with a randomly ordered sequence for the remaining 300 epochs. **Please note the high decrease of the error in the middle of the plot - i.e. when the *model IV* training sequence was replaced by the random one.**

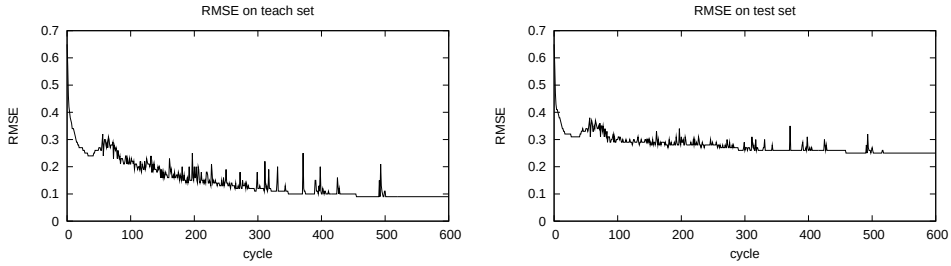


Fig. 2. RMSE obtained in each cycle using randomly ordered training data

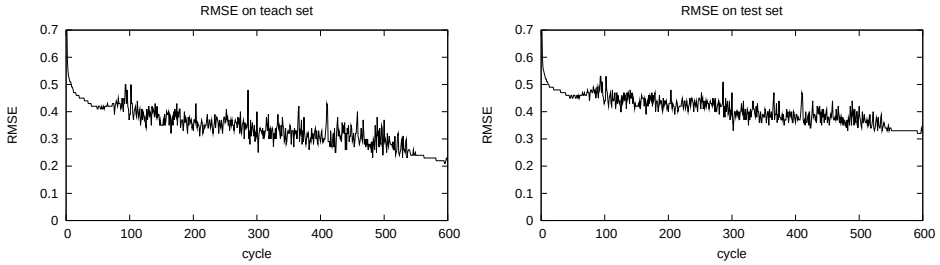


Fig. 3. RMSE when using training data ordered according to *Model II*

Following the idea of switchings sequences applied in the previous experiment a simulation of the training process with more frequent sequence exchange was performed. In this case the sequence ordered according to *model II* was exchanged with the random sequence after every 20 training epochs. The results in terms of RMSE plot are presented in Fig. 5. A comparison of RMSE in the above case with a pure random case is presented in Fig. 6. It is remarkable that **after each alteration of *model II* sequence with a random one RMSE becomes lower than in pure random case**. The possible explanation is that non-uniformity of elements' distribution has the effect in local changes of weights' change direction during presentation of training sequences which consequently allows the network to escape from local shallow minima.

2.2 Proposition of training sequence switching

Due to observed activity of ordered sequences it should be considered to interleave them with random ones in the training process. It is therefore proposed to apply a model with decreasing probability of using ordered sequences in the training process. Let

$$P(t) = pe^{-\eta t} \tag{4}$$

be the probability of presenting ordered sequence, where t is the number of the training epoch, p - the initial probability, η - positive coefficient of probability decrease. Having two training sequences - one ordered according to any of the above described four

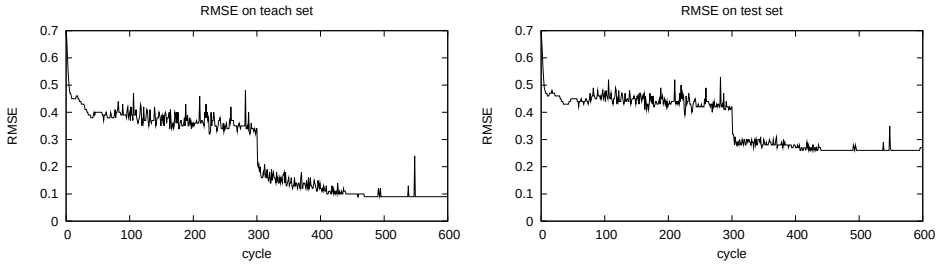


Fig. 4. RMSE in case the training data ordered according to *Model IV* is used in the first 300 epochs followed by training with the random sequence in the remaining 300 epochs

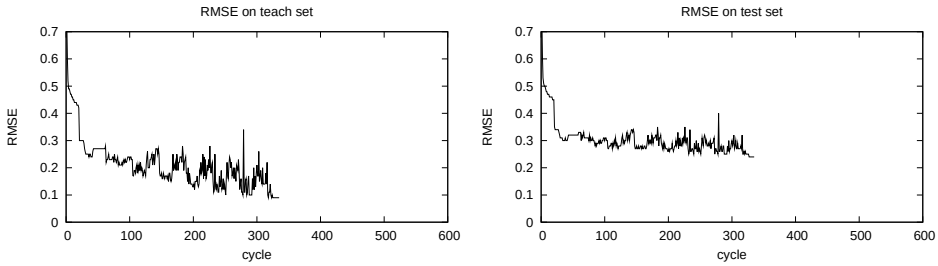


Fig. 5. RMSE in case when training data ordered according to *Model II* is periodically (after every 20 cycles) exchanged with a random sequence

models and the other one being a purely random) in each epoch the ordered training sequence is chosen according to the above probability. Since the remainder of the paper will be devoted to the proposed algorithm, henceforth, *model I*, *model II*, *model III* and *model IV* will refer to the above training method in which the respectively ordered sequence is interleaved with the random one. As a special case also two randomly chosen (fixed) sequences are considered as the two interleaved sequences. This case will be denoted by *switched random*.

3 Performance of proposed algorithm

In each case training process consisted of 600 epochs, initial probability p was equal to 1.0 and η was chosen so that $P(600) = 0.03$.

3.1 Independent training

Statistics (mean RMSEs and Standard Deviations) of populations of neural networks obtained by training with given model of an order are presented in Table 1. Sequences are ordered according to RMSE values on the test set. Visualization of the RMSE values is presented Figure 4, in which all populations are presented. Each dot represents one neural network. Initial weights of these networks were independently chosen at random.

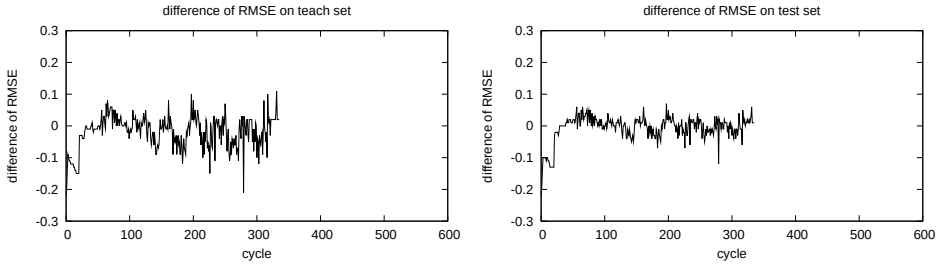


Fig. 6. Difference between RMSEs calculated in Fig. 2 (pure random case) and Fig. 5 (*model II* interleaved with random sequence).

Table 1. Statistics of RMSE

model	*mean RMSE on train set*	*SD RMSE on train set*	*mean RMSE on test set*	*SD RMSE on test set*
model III	0.0798	0.0140	0.2591	0.0109
model I	0.0844	0.0115	0.2602	0.0116
model IV	0.0818	0.0138	0.2621	0.0098
model II	0.0841	0.0206	0.2640	0.0128
switched random	0.0882	0.0244	0.2640	0.0165
pure random	0.0939	0.0209	0.2668	0.0118

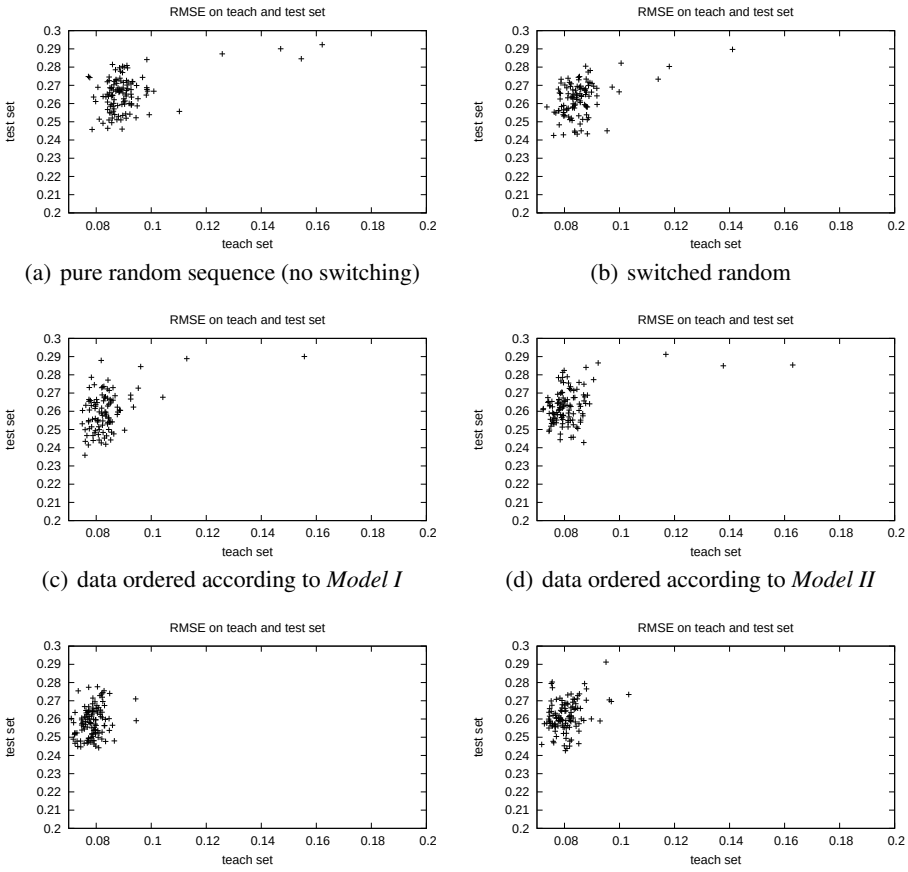
It is remarkable that **among populations obtained with use of randomly ordered sequences and ones obtained using sequences ordered according to proposed models exists a statistically significant difference**. P-values for hypothesis about significant difference (obtained from t-Student test) are presented in Table 2.

Table 2. P-value of hypothesis that distributions of RMSE on the training set are different.

	model III	*switched random*	*model IV*	*model II*	*model I*	*pure random*
model III	1					
switched random	0.002	1				
model IV	0.288	0.0170	1			
model II	0.069	0.1688	0.3256	1		
model I	0.009	0.1449	0.130	0.874	1	
pure random	0.000	0.0613	0.000	0.000	0.000	1

It can be concluded that an improvement of average RMSE in the best case of randomly ordered sequence and the best case of the ordered one (*model III* vs *switched random*) is equal to 9.52% and 1.84%, resp. on the training and tests sets.

The average pattern classification result of the best model (*model III*) on the test set was equal to 92.55%.



(a) pure random sequence (no switching)

(b) switched random

(c) data ordered according to *Model I*

(d) data ordered according to *Model II*

(e) data ordered according to *Model III*

(f) data ordered according to *Model IV*

Fig. 7. RMSE on training and test sets in last epoch in case of using ordered sequence switched by random one according to formula (4).

3.2 Training represented as dependent variables

In order to eliminate randomness of neural network initial weights (which can be considered as a noise in case of independent samples) the research of dependent samples was performed. Population consisted of 64 neural networks and each of them has been trained 6 times (once for each training model) - each time with the same set of initial weights.

In order to analyze the influence of ordering on possibility of obtaining a network with good generalization abilities the top 20 recognition results on the test set were selected. The average and the maximum pattern classification results on the test set of these networks were equal to 93.93% and 94.49%, resp. Fractions of networks trained according to particular models' orders, which belonged to this group are presented in Table 3. Note, that **reverted models are dominating (**70%**)** and also **no neural network trained exclusively with random sequences has qualified to the set**.

Table 3. Percentage of sequences among the top 20 networks on the test set.

	model III	*switched random*	*model IV*	*model II*	*model I*	*pure random*
percentage	25%	0%	45%	10%	20%	0%

4 Conclusions

The problem of ordering training patterns is essential in supervised learning with neural networks. In the paper a new method of ordering training patterns is proposed and experimentally evaluated. It was shown that proposed approach produces in average better results than training without its use in the sample problem representing clustered pattern space. Some theoretical considerations supporting this result has been provided.

Tests in other problem domains are under research. Other possible uses of ordered sequences (e.g. as a measure of generalization ability of network architecture) are considered as future research plans.

Acknowledgment

This work was supported by the Warsaw University of Technology under grant no. 504G 1120 0008 000. Computations were performed using grid provided by Enabling Grids for E-sciencE (EGEE) project.

References

1. Ratcliff, R.: Connectionist Models of Recognition Memory: Constraints Imposed by Learning and Forgetting Functions. Psychological Review, 97(2), (1990) 285–308
2. French, R. M.: Catastrophic Forgetting in Connectionist Networks. Trends in Cognitive Sciences, 3(4), (1999) 128–135.)
3. de Carvalho, A., Brizzotti M. M. : Combining RBF Networks Trained by Different Clustering Techniques
   Neural Processing Letters 14, (2001) 227–240
4. Mańdziuk, J., Shastri, L.: Incremental Class Learning approach and its application to Handwritten Digit Recognition. Information Science, 141(3-4), (2002) 193–217