ELSEVIER

Pattern Recognition Letters 17 (1996) 819–823

Pattern Recognition Letters

A sample set condensation algorithm for the class sensitive artificial neural network

C.H. Chen [a], Adam Jóźwik [b,*]

[a] University of Massachusetts Dartmouth, Electrical and Computer Engineering Dept., 285 Old Westport Road, N. Dartmouth, MA 02747, USA

[b] Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Sciences, Trojdena 4, PL 02-109 Warsaw, Poland

Received 26 July 1995; revised 4 March 1996

Abstract

The paper presents a new way to reduce the size of the training set without significantly decreasing the classification quality. The effectiveness of the proposed algorithm is examined for the class sensitive neural network (CSNN) presented in WCNN93 by Chen and You (1993) although the same approach can be applied also to other kinds of classifiers. Ten different experiments with a very large remote sensing data set were performed to verify the proposed approach.

Keywords: Statistical pattern recognition; Artificial neural networks; Training set condensation

1. Introduction

The training time of an artificial neural network depends both on the number of features used and on the size of the training set. The number of features can be reduced in a feature selection process that requires a certain number of training sessions. We have in mind the forward and the backward feature selection strategies described by Devijver and Kittler (1982). Thus, it seems that the reduction of the training set size could remarkably accelerate the training process and therefore also the feature selection.

The algorithm for the training set size reduction, we propose below, requires calculation of some distances between two objects. We have decided to use the Euclidean measure. The objects represented in the training set may be described by different units. For this reason the data ought to be standardized. The standardization will be used only to find the training set of the reduced size. However, the training set of the reduced size will finally contain the original nonstandardized data. For this purpose the following equation can be used:

$$x[i,j] := (x[i,j] - mv[j])/sd[j],$$

where i identifies an object, j is the feature number, $mv[j]$ is the mean value of the jth feature, and $sd[j]$ its standard deviation. The values $mv[j]$ and $sd[j]$ will be derived from the training set. We will use this equation as it gives an equal "weight" to each feature.

The idea of the proposed algorithm consists in the division of the training set into some subsets with use of the standardized data. Next, these subsets are

* Corresponding author.

0167-8655/96/$12.00 © 1996 Elsevier Science B.V. All rights reserved

PII S0167-8655(96)00041-4

replaced by their gravity centers calculated for the original nonstandardized data. These gravity centers assume the same labels as the majority of points of the corresponding subsets. Ties can be broken by the class most heavily represented in the training set.

There are no formal methods reported in the literature on sample set condensation. The procedures reported are based on cluster analysis (see e.g. Devijver and Kittler, 1982) which has no control of the resulting number of condensed samples. Furthermore the proposed algorithm is computationally simpler.

2. A brief description of CSNN

The class-sensitive neural network is a feedforward artificial neural network which is particularly suitable for multi-class pattern classification. Each class is represented by a subnet. All subnets which share the same input are otherwise uncoupled. A typical network structure for a two-dimensional input vector and three classes, and with one hidden layer is shown in Fig. 1. The weight adjustments are made such that weight reinforcement is applied to the correct pattern class of the input vector. The control strategy makes use of both back-propagation and the correlations between the real and desired outputs. Experiments with artificial and real data show that the CSNN performs consistently better than the popular back-propagation trained neural networks by 6 to 10% in recognition rate. The condensed samples are used as the input vectors to the CSNN for classification.

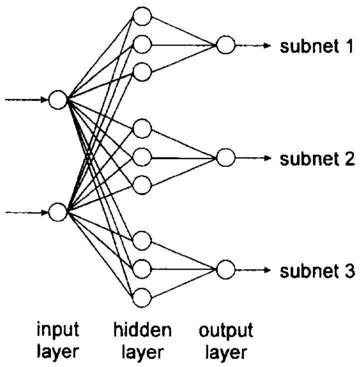

Fig. 1. CSNN structure.

3. Training set condensation algorithm

To describe our algorithm we introduce the term "diameter" of the set. As the diameter of the training set (or of a subset of the training set) we will assume the distance between its two farthest points.

First we will show how the proposed algorithm operates. A one-dimensional example, presented in Fig. 2, seems to be sufficient for this purpose.

The training set consists of 9 points, 5 from class 1 and 4 from class 2. Its two farthest points have coordinates 1 and 37. The first division corresponds to the straight line that passes through the point 19, i.e. in the middle between points 1 and 37. Now we have two subsets. The next division is performed for the subset that contains a mixture of points from two classes. If more than one subset satisfies this condition, then we divide the subset with the largest diameter. So, the left subset {1, 4, 7, 13, 17} will be divided in the second step. The subset {23, 27, 35, 37} will be divided in the third. After the next two steps the fifth division will be performed. In this situation no subset contains a mixture of points. There are not two points from the different classes in the same subset. The mixed subsets have been exhausted. Further division can be realized now only for uniform (nonmixed) subsets. For instance, the sixth division would pass through the point 4. The point 4 can join to the left as well as to the right subset.

Fig. 2 shows the compressed training set after five steps. The compressed training set is created by gravity centers of the obtained subsets.

Below, we describe the proposed algorithm more precisely.

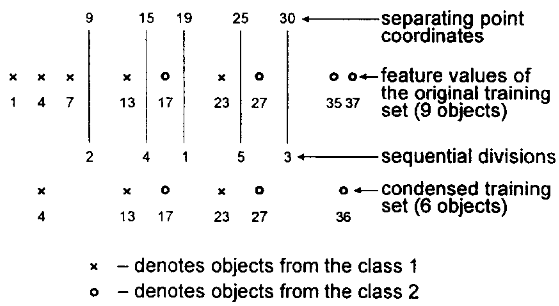

Fig. 2. A one-dimensional example for a two-class problem.

Algorithm

1. Select the desidered number n_d of objects (points) in the reduced training set;
2. Put $n_c := 1$; (" := " means the same as in the Pascal language, n_c is the current number of subsets of the training set)
3. Assume $D :=$ training set, $C(1) := D$ and $i := 1$;
4. Find two farthest points P_1 and P_2 in the set D;
5. Divide the set D into two subsets D_1 and D_2, where
   $D_1 := \{P \in D\colon d(P, P_1) \leqslant d(P, P_2)\}$,
   $D_2 := \{P \in D\colon d(P, P_2) < d(P, P_1)\}$;
6. $n_c := n_c + 1$, $C(i) := D_1$, $C(n_c) := D_2$;
7. Put $I_1 := \{i\colon C(i)$ contains objects from two classes at least$\}$, $I_2 := \{i\colon i \leqslant n_c\} - I_1$;
8. Put $I := I_1$ if I_1 is non-empty else $I := I_2$;
9. Find the pair of two farthest points $Q_1(i)$ and $Q_2(i)$ in each $C(i)$ for $i \in I$;
10. Find j such that $d(Q_1(j), Q_2(j)) = \max d(Q_1(i), Q_2(i))$ for $i \in I$, i.e. find a set $C(j)$ with largest diameter;
11. Put $D := C(j)$, $P_1 := Q_1(j)$ and $P_2 := Q_2(j)$;
12. If $n_c < n_d$ then go to 5;
13. Find gravity centers $G(i)$ for each $C(i)$, $i = 1, 2, \ldots, n_d$;
14. Assign to each $G(i)$ the class that is most heavily represented in $C(i)$, ties break by largest class and further randomly. The points $G(i)$, $i = 1, 2, \ldots, n_d$, form the condensed training set.

4. Data set description

The above-presented algorithm was applied to remote-sensing images of the multisensorial type. Particularly, we considered images acquired by two sensors installed on an aircraft: a Daedalus 1268 Airborne Thematic Mapper (ATM) scanner, and a PLC band, fully polarimetric, NASA/JPL airborne imaging radar system. The flights took place in July and August 1989, respectively. The geographical location was the Feltwell area. The average registration error was about 1 pixel.

The registered ATM image was filtered by a linear smoothing and context-sensitive enhancement filter; then it was segmented by a multiband region-growing technique (Serpico and Roli, 1995). We selected regions belonging to the following classes: carrots, potatoes, stubble, sugar beet and wheat. For each region, a set of 15 features was computed, including original optical and radar channels, texture features calculated by using the above channels and combinations of optical bands.

The following is a brief feature description.

Features 1, 2, 3, 4, 5, 6 – the responses of the optical sensor, i.e. the Daedalus sensor, for band 2, band 3, etc.

Feature 7 – for the C band with HH polarization.

Feature 8 – response of the radar sensor for the L band with HV polarization.

Feature 9 – response of the radar sensor for the P band with VV polarization.

Feature 10 – Mandelbrot texture for the C polarization HH.

Feature 11 – Mandelbrot texture for the band C polarization HV.

Feature 12 – Mandelbrot polarization for the band P polarization HV.

Feature 13 – synthetic feature computed as ratio between band 7 and (band 5 + band 7 + band 9).

Feature 14 – synthetic feature computed as ratio between band 9 and (band 5 + band 7 + band 9).

Feature 15 – synthetic feature computed as ratio between band 7 and (band 3 + band 5 + band 7).

More detailed information concerning the data set can be found in the paper of Serpico and Roli (1995).

5. Results of experiments

We have considered a data set that contained 8839 objects, 15 features and 5 classes. Next, we randomly selected 2440 objects to use as the training set, and the set of the remaining 6399 objects was treated as the test set. Such an experiment was repeated ten times. In each experiment the training set was condensed at first into a set with 38 objects, secondly into a set with 76 objects, then into sets with 152, 305, 610 objects and finally into a set that contained 1220 objects. For each of the ten experiments the CSNN was trained seven times: for the original set of 2440 objects and the six condensed sets with 1220, 610, 305, 152, 76 and 38 objects in each set. The condensed sets with 38, 76, 152, 305,

Table 1
Results of 10 experiments with the condensed training sets

Exp. no.	Number of objects in training respectively test sets						
	2440 6399	1220 6399	610 6399	305 6399	152 6399	76 6399	38 6399
1	91.78	92.01	89.97	89.03	84.72	67.26	63.42
2	94.73	93.25	91.95	89.97	84.92	63.13	59.52
3	93.45	91.23	89.61	89.40	82.00	70.71	54.34
4	92.86	91.37	90.25	91.03	86.00	53.29	51.85
5	94.23	92.76	93.23	88.86	83.70	54.59	53.13
6	94.12	92.78	89.31	91.47	83.28	75.84	76.14
7	93.08	92.23	93.56	90.70	82.43	73.54	68.79
8	89.30	91.67	92.66	89.83	80.47	81.08	74.09
9	95.05	94.81	86.58	88.53	73.09	67.20	54.60
10	94.16	91.26	90.47	89.45	74.03	66.67	63.13
Average	93.28	92.34	90.76	89.83	81.46	67.33	61.90

610 and 1220 objects were found in the same computational process since the proposed algorithm produces first the set with 38 objects, then, by continuing it, the sets with 76, 152, 305, 610 objects and, finally, by further continuation, the set with 1220 objects. To create the first set of 38 objects 7 minutes were necessary and an additional 5 minutes were sufficient to form the remaining five sets with 38, 76, 152, 305, 610 and 1220 objects. For computations an IBM compatible PC 486/66 MHz was used.

Table 1 contains the results for each of the ten experiments with seven different training sets, the original one and the six condensed training sets.

The CPU time for 1220 objects in the training set is 85% of that with 2440 objects in the training set, for 610 objects and 305 objects in the training set the CPU time is 80% and 60% respectively of that with 2440 objects in the training set. Note that with the training sample size reduced by a factor of 8, there is only a 4% decrease in the average recognition rate.

It is worth to compare the results obtained for the condensed training sets and for the training sets of the same size randomly chosen from the original sets of 2440 objects. To avoid unnecessary computations we have constrained our experiments to the size of 305 objects in each of the sets. The results are presented in Table 2.

We see that the condensed training set offers a better classification quality than the training set of the same size but randomly chosen from the original set of 2440 objects.

Comparing the results in Table 1 and those given in Table 2 we notice that condensation from a 2440-point to a 305-point training set leads to only 3.5% loss of accuracy, while a random choice of 305 points in the training set causes on average 19.4% loss in accuracy.

The same data have been used by Serpico and Roli (1995). They have considered another type of

Table 2
Results of 10 experiments for the condensed and randomly chosen training sets with 305 objects

Exp. no.	No. of objects in training respectively test sets	
	condensed 305 6399	randomly chosen 305 6399
1	89.03	84.23
2	89.97	72.70
3	89.40	88.11
4	91.03	73.14
5	88.86	84.23
6	91.47	71.64
7	90.70	71.95
8	89.83	70.37
9	88.53	59.35
10	89.45	62.81
Average	89.83	73.85

artificial neural network and the k-NN classifier. The error rates were 86.46 and 89.85 percent, respectively. So, the CSNN used by us offers a similar classification quality. However, our main interest was sample set condensation and the relation between the size of the condensed training set and classification quality.

6. Concluding remarks

Sample condensation or reduction is particularly necessary when the data set involved is large such as in the remote sensing image recognition problems. A new algorithm is proposed in this paper and shown to preserve the classification accuracy quite well even when the training sample size reduction is by a factor of 8.

Acknowledgements

The authors wish to thank the support of NATO Collaborative Research Grant. They also thank Hunting Technical Services Ltd for providing images and Dr. Sebastiano Serpico of the University of Genova for giving us the ready data set with extracted features. Thanks also go to graduate student Yongjiang Yu of Umass Dartmouth for computer study.

References

Chen, C.H. and G.H. You (1993). Class-sensitive neural network. Presented at *WCNN93*, appeared also in *Neural, Parallel & Scientific Computations* 1, 93–96.

Devijver, P.A. and J. Kittler (1982). *Pattern Recognition: A Statistical Approach.* Prentice-Hall, Englewood Cliffs, NJ.

Serpico, S.B. and F. Roli (1995). Classification of multisensor remote sensing images by structured neural networks. *IEEE Trans. Geoscience Remote Sensing* 33 (3), 562–578.