Probabilistic Neural Network Structure Reduction for Medical Data Classification

Maciej Kusy and Jacek Kluska

Faculty of Electrical and Computer Engineering, Rzeszow University of Technology,
35-959 Rzeszow, W. Pola 2, Poland
{mkusy,jacklu}@prz.edu.pl

Abstract. Probabilistic neural network (PNN) consists of the number of pattern neurons that equals the cardinality of the data set. The model design is therefore complex for large database classification problems. In this article, two effective PNN reduction procedures are introduced. In the first approach, the PNN's pattern layer neurons are reduced by means of a k-means clustering procedure. The second method uses a support vector machines algorithm to select pattern layer nodes. Modified PNN networks are compared with the original model in medical data classification problems. The prediction ability expressed in terms of the 20% test set error for the networks is assessed. By means of the experiments, it is shown that the appropriate pruning of the pattern layer neurons in the PNN enhances the performance of the classifier.

Keywords: probabilistic neural network, k-means clustering, support vector machines, classification, prediction ability.

1 Introduction

The probabilistic neural network (PNN), proposed by Specht [1] is a direct implementation of the Bayes classifier. It can quickly learn from input data but requires one neuron in the pattern layer for each training example [2]. PNNs have found their implementation in variety of classification fields. It was presented in image classification and recognition [3], [4], earthquake magnitude prediction [5] or medical diagnosis and prediction [6]–[9]. The important contribution was provided in [10] where PNN was applied to pattern classification in time-varying environment.

Since PNN consists of a single node for each data, various modifications of the network have been proposed. For example, in [11], by estimating probability density functions as a mixture of the Gaussian densities with varying covariance matrices it was possible to design PNN so that it used fewer nodes than training patterns. The work in [12] presented learning vector quantization technique for finding representative patterns to be used as neurons in PNN. In [13], the authors presented a Generalized Fisher algorithm for PNN and showed that it required significantly fewer nodes and interconnection weights than original model. The reference in [3] presented the reduction of the size of the training data for PNN

L. Rutkowski et al. (Eds.): ICAISC 2013, Part I, LNAI 7894, pp. 118–129, 2013.

by hierarchical clustering. Here, the reciprocal neighbors technique was applied which allowed to gather the examples which were closest to each other. In [14], the quantization method for PNN structure was proposed. The input space was divided into a fixed-size hyper-grid and within each hyper-cube a representative cluster centres were computed. Therefore, the number of training vectors in each hyper-cube was reduced to one. The research in [15] presented the automatic construction of PNN by the use of a dynamic decay adjustment algorithm. The model was dynamically built during training which automatically optimized the number of hidden neurons. The work reported in [4] proposed PNN with no distance matrix needed for storing the pairwise distances between input examples and the vector to be classified. It was achieved by maintaining the nearest neighbor table of indices of the nearest cluster for each cluster. In [16], a supervised PNN structure determination algorithm was introduced. The procedure employed genetic algorithm for pattern layer neuron selection.

It is necessary to note that the PNN model is equipped with the intrinsic smoothing parameter of the pattern layer neurons activated by Gaussian function. It must be estimated on the basis of a classification performance. Three approaches are usually regarded: single parameter for whole PNN, separate parameter for each variable (dimension) or single parameter for each class. In the research, a diverse procedures have been developed to solve the problem [2], [6], [16], [17].

In this article, two alternative approaches of the structure minimization of the probabilistic neural network are introduced. The first method is based on the application of k-means clustering algorithm to input data in order to determine the optimal number of centroids as the representation of the pattern layer neurons. In the second solution, the support vector machine procedure is applied which, out of the entire training database, provides the set of support vectors. The support vectors form then the layer of pattern nodes of PNN. Both techniques are tested on the medical data sets.

This paper is composed of the following sections. Section 2 discusses probabilistic neural network highlighting its basics, a structure and a principle of operations. In Section 3, the reduction of PNN structure by means of a k-means clustering and support vector machines algorithm is proposed. Section 4 briefly describes the input data used in the research. Additionally, the performance of the standard and the modified PNN models is here verified. Finally, in Section 5, the conclusions are presented.

2 Probabilistic Neural Network

Assume, we are given an input vector $\mathbf{x} \in \mathbb{R}^n$ which belongs to one of the predefined classes $g = 1, 2, \ldots, G$. Let the probability of the vector $\mathbf{x}$ belonging to the class g equals p_g, the cost associated with classifying the vector into class g is c_g and that the probability density functions: $y_1(\mathbf{x}), y_2(\mathbf{x}), \ldots, y_G(\mathbf{x})$ for all classes are known. Then, according to the Bayes theorem, when $g \neq h$, the vector $\mathbf{x}$ is classified to the class g, if $p_g c_g y_g(\mathbf{x}) > p_h c_h y_h(\mathbf{x})$. Usually $p_g = p_h$ and $c_g = c_h$, thus one can infer that if $y_g(\mathbf{x}) > y_h(\mathbf{x})$, then the vector $\mathbf{x}$ belongs to the class g.

In real data classification problems, data set distribution is usually unknown and an approximation of the probability density function $y_g(\mathbf{x})$ has to be determined. This can be achieved using the Parzen method [18] where the probability density function for multiple variables can be expressed as follows

$$y\left(\mathbf{x}\right) = \frac{1}{l}\sum_{i=1}^{l} W_i\left(\mathbf{x}, \mathbf{x}_i\right), \tag{1}$$

where $W_i\left(\mathbf{x}, \mathbf{x}_i\right) = \sigma_1^{-1} \ldots \sigma_n^{-1} F\left(\sigma_1^{-1}\left(x_{i1} - x_1\right), \ldots, \sigma_n^{-1}\left(x_{in} - x_n\right)\right)$, $F\left(\cdot\right)$ is the weighting function which has to be appropriately selected [19], l is the number of input patterns, and $\sigma_1, \ldots, \sigma_n$ denote standard deviations computed relative to the mean of n variables $x_1, \ldots, x_n$. Usually, the Gaussian function is a common choice for weighting in (1).

The formula in (1) defines the structure and the operation of PNN. If we consider a Gaussian function as the activation for the probability density function and assume that this function is computed for the examples of class g then Parzen's definition takes the following form

$$y_g\left(\mathbf{x}\right) = \frac{1}{l_g\left(2\pi\right)^{n/2}\left(\det \Sigma_g\right)^{1/2}} \sum_{i=1}^{l_g} \exp\left(-\frac{1}{2}\left(\mathbf{x}_{g,i} - \mathbf{x}\right)^T \Sigma_g^{-1}\left(\mathbf{x}_{g,i} - \mathbf{x}\right)\right), \tag{2}$$

where $\Sigma_g = \mathrm{diag}\left(\sigma_{g,1}^2, \ldots, \sigma_{g,n}^2\right)$ is the covariance matrix, l_g is the number of the examples of class g, $\sigma_{g,j}$ denotes the smoothing parameter associated with j-th variable and the g-th class, and $\mathbf{x}_{g,i}$ is the i-th training vector $(i = 1, \ldots, l_g)$ from the class g. The formula presented in (2) provides one of $g = 1, \ldots, G$ summation neurons of PNN structure. The elements of the preceding layer, the pattern neurons, feed the component to the sum which is measured over each of the examples of g-th class. Therefore, l_g hidden neurons constitute the input for g-th summation neuron. Finally, the output layer determines the output for the vector $\mathbf{x}$ in accordance with the Bayes's decision rule based on the outputs of all the summation layer neurons

$$G^*\left(\mathbf{x}\right) = \arg\max_g \left\{y_g\left(\mathbf{x}\right)\right\}, \tag{3}$$

where $G^*\left(\mathbf{x}\right)$ denotes the predicted class for the pattern $\mathbf{x}$. Thus, the pattern layer requires $l = l_1 + \ldots + l_G$ nodes.

In this paper, single smoothing parameter for each attribute and class is applied. The choice of this variant of σ selection imposes, in accordance with formula (2), the inevitability of storing a $G \times n$ matrix of σ's. Hence, the g-th summation neuron yields to the decision layer the output signal (2) but with $\sigma_{g,j}$ as the intrinsic parameter. Therefore, the smoothing parameter is computed for the j-th variable of each class g. Such an approach gives the possibility of

emphasizing the similarity of the vectors belonging to the same class. The values of σ's are determined using the conjugate gradient method.

3 PNN Reduction Methods

In this section, two approaches of PNN structure simplification are presented. Both solutions consist in decreasing the number of pattern neurons of the network. The first method is based upon k-means procedure. The second idea utilizes the support vectors for PNN training.

3.1 The Use of k-Means Clustering for PNN Structure Reduction

The k-means algorithm is a data clustering method [20] which is considered to be one of the top ten algorithms in data mining [21]. The procedure finds k clusters, such that all the records within each cluster are similar to each other and distinct from records in other clusters. The grouping process relies on the iterative minimization of the sum of squared distances computed between input vectors and the cluster center. An initial set of clusters is defined, and the cluster centres are repeatedly updated until no modification of their coordinate values takes place.

The first approach in PNN structure reduction uses k-means algorithm in a simple iterative way. The number of clusters of class g in s-th iteration is determined according to the formula

$$i_{s,g} = \text{round}\left(\frac{s}{N}\eta l_g\right), \quad s = 1, \ldots, N-1, \tag{4}$$

where η is a fraction of training data ($1-\eta$ is the part for testing) and round (x) is the function that rounds the real positive number x to the nearest integer. In this paper, we assume $\eta = 0.8$, and $N = 10$. It is important to notice, that only $i_{s,g}$ pattern layer neurons of class g are involved in computation of the signal for the summation layer neuron. The **Algorithm 1** summarizes the proposed method.

Algorithm 1. PNN architecture optimization based on k-means clustering.

```
Randomly determine training and test sets
for s = 1 to N − 1 do
    for g = 1 to G do
        Compute i_{s,g} cluster centres for training set
    end
    Train PNN on c_s = Σ_{g=1}^{G} i_{s,g} cluster centres
    Read σ_1^(1), ..., σ_n^(G) which minimize PNN training error
    Calculate test error E_test for PNN on test set
end
```

Now, if we define the reduction ratio R as a quotient of the number of pattern neurons by the size of the training data set for the PNN

$$R(s) = \frac{\sum_{g=1}^{G} i_{s,g}}{\eta l} \cong \frac{s}{N}, \quad s = 1, \ldots, N-1, \tag{5}$$

then the optimal ratio in the sense of the stated problem is $R(s^*)$ for

$$s^* = \arg \min_{1 \leqslant s \leqslant N-1} E_{test}(s), \tag{6}$$

where $E_{test}(s)$ is the test error obtained by the s-th cluster's variant and s^* is computed numerically.

3.2 The Use of Support Vector Machines for PNN Structure Reduction

Support vector machine (SVM) [22] is one of the most accurate methods among all well-known classification algorithms [21]. It constructs an optimal classifier for the input vector $\mathbf{x}_i$ $(i = 1, \ldots, l)$ from the class labelled $y_i = \pm 1$. Two types of SVM algorithms are usually applied in data mining problems: C-SVM model and ν-SVM model [23]. In this research, C-based SVM is used. The C-SVM training amounts to the solution of the following quadratic programming optimization (QP) problem

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} W(\boldsymbol{\alpha}) = -\dfrac{1}{2}\langle \boldsymbol{\alpha}, \mathbf{H}\boldsymbol{\alpha} \rangle + \langle \boldsymbol{\alpha}, \mathbf{1} \rangle \\ \mathbf{0} \leqslant \boldsymbol{\alpha} \leqslant C \cdot \mathbf{1}, \quad \langle \boldsymbol{\alpha}, \mathbf{y} \rangle = 0, \end{cases} \tag{7}$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_l]^T$ is the vector of Lagrange multipliers, $\mathbf{H} = \{y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)\}$ is $l \times l$ matrix, $K(\cdot, \cdot)$ is the kernel function, $\mathbf{y} = [y_1, \ldots, y_l]^T$ is the vector of class labels, $\mathbf{0} = [0, \ldots, 0]^T$ and $\mathbf{1} = [1, \ldots, 1]^T$. Once the solution of (7) is obtained in terms of $\boldsymbol{\alpha}$ vector, the optimal classifier is formulated

$$\text{class}(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right). \tag{8}$$

The input vectors $\mathbf{x}_i$ having $\alpha_i > 0$ are called support vectors (SVs). They constitute a sufficient sub-set out of given input data for a sample prediction. As it can be observed, the solution of the QP problem in (7) involves the constraint for α_i which requires the choice of unknown C parameter. The value of C coefficient introduces additional capacity control for the classifier. The adjustment of C provides greater or smaller number of support vectors what, in turn, influences the classification accuracy. In this research, by setting different values of C constraint, we are capable of obtaining different sets of support vectors. Depending on the considered data set and the value of C, the number of pattern neurons of PNN changes.

The final classification outcome also depends on the kernel function $K(\cdot,\cdot)$ applied in (8). Much study in recent years has been devoted to adopting different kernels for SVM [24]–[26]. In this contribution, the Gaussian kernel function is applied with the spread constant (sc) as the parameter:

$$K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\,(sc)^2}\right). \tag{9}$$

An appropriate range of the spread constant has to be estimated which is realized numerically with the assumption of achieving the highest generalization ability of the classifier.

For C and sc parameters, the final sets of values A_C and A_{sc} are assumed, respectively. The SVM based PNN reduction methodology is summarized in form of **Algorithm 2**.

Algorithm 2. PNN architecture optimization based on SVM

```
Randomly determine training and test sets
for C ∈ A_C and sc ∈ A_sc do
    Perform SVM classification on training set
    Select support vectors SVs
    Train PNN on SVs
    Read σ_1^(1), ..., σ_n^(G) which minimize PNN training error
    Calculate test error E_test for PNN on test set
end
```

The particular values for both, A_C and A_{sc} are provided in Section 4.2.

4 Results

In the simulations, seven UCI machine learning repository medical data sets are used [27]: Wisconsin breast cancer (WBC): 683 instances with 9 attributes (binary classification), Pima Indians diabetes (PID): 768 cases having 8 features (binary classification), Haberman's survival (HS): 306 patients and 3 measured variables (binary classification), Cardiotocography (CTG): 2126 measurements on 23 attributes (three state classification), Thyroid (T): 7200 instances with 21 attributes (three state classification), Dermatology (D): 358 cases each of 34 features (six data sets classification) and diagnostic Wisconsin breast cancer (DWBC): 569 instances having 30 variables (binary classification). Additionally, authors' real ovarian cancer (OC) data set is used in the simulations: it represents 199 women after ovarian cancer treatment with 17 parameters registered for each case. The data is obtained from the Clinical Department of Obstetrics and Gynecology of Rzeszow State Hospital in Poland. The analysis of treatment of ovarian cancer and its hormonal and genetic aspects are studied in [28].

In all cases, no data preprocessing (normalization, feature extraction) is performed. After random selection of 20% of the input data for testing purposes, training and test sets are preserved for all the data and each method what makes both approaches comparable.

The following sections present the comparison of the prediction ability measured for the standard PNN model and the networks for which the number of pattern neurons is reduced by means of two proposed approaches: the application of k-means method to cluster the data, and the use of the support vectors as the new database. The prediction ability of the examined classifiers is determined on the basis of the test error (E_{test}) computed on the 20% of input examples randomly extracted from each of given data sets. The number of the test vectors for WBC, PID, HS, CTG, T, D, DWBC and OC data sets is equal 137, 154, 61, 425, 1440, 72, 456 and 40, respectively. The bottom rows of Tables 1–4 present the test error for original PNN.

4.1 Experimental Results after the Use of Algorithm 1

Tables 1–4 illustrate the test error computed after data clustering according to **Algorithm 1** for all considered data sets. The sum $\sum_{g=1}^{G} i_{s,g}$ defines the total number of pattern layer neurons. It can be observed that in each data classification case, by reducing the number of pattern neurons of PNN, it is possible to find the smaller test error than the one computed with the use of all pattern neurons of the model. It is also worth to note that, in all data classification cases, the decrease of the test error takes place more than once.

The most gainful reduction ratio R can be read from the Tables 1–4, e.g. for WBC data set it takes the value $R = 55/(0.8 * 683) \cong 0.1$. Therefore, instead of 683 cases of original data we can use their substitutes but about 10 times smaller in number.

Table 1. The number of cluster centres used in determining PNN structure and the test error for WBC data set (left table) and PID data set (right table)

s	$i_{s,1}$	$i_{s,2}$	Pattern neurons	E_{test} [%]
1	36	19	55	6.569
2	71	38	109	8.029
3	107	57	164	10.949
4	142	76	218	15.328
5	178	96	274	14.599
6	213	115	328	10.219
7	248	134	384	9.489
8	284	153	437	11.679
9	320	172	492	7.299
All	355	191	546	9.489

s	$i_{s,1}$	$i_{s,2}$	Pattern neurons	E_{test} [%]
1	40	21	61	62.337
2	80	43	123	29.220
3	120	64	184	29.220
4	160	86	246	38.311
5	200	107	307	31.168
6	240	128	368	28.571
7	280	150	430	29.220
8	320	171	491	33.766
9	360	193	553	32.467
All	400	214	614	31.818

Table 2. The number of cluster centres used in determining PNN structure and the test error for HS data set (left table) and CTG data set (right table)

s	$i_{s,1}$	$i_{s,2}$	Pattern neurons	E_{test} [%]
1	18	7	25	24.590
2	36	13	49	24.590
3	54	20	74	26.229
4	72	26	98	27.868
5	90	33	123	26.229
6	108	39	147	26.229
7	126	46	172	26.229
8	144	52	196	27.868
9	162	59	221	57.377
All	180	65	245	31.147

s	$i_{s,1}$	$i_{s,2}$	$i_{s,3}$	Pattern neurons	E_{test} [%]
1	132	24	14	170	19.058
2	265	47	28	340	21.411
3	397	71	42	510	28.470
4	530	94	56	680	36.941
5	662	118	71	851	35.294
6	794	142	85	1021	11.529
7	927	165	99	1191	15.529
8	1059	189	113	1361	12.470
9	1191	212	127	1530	16.941
All	1324	236	141	1701	15.529

Table 3. The number of cluster centres used in determining PNN structure and the test error for T data set (left table) and D data set (right table)

s	$i_{s,1}$	$i_{s,2}$	$i_{s,3}$	Pattern neurons	E_{test} [%]
1	13	29	533	575	60.625
2	27	59	1067	1153	43.402
3	40	88	1600	1728	89.166
4	53	118	2133	2304	7.222
5	67	147	2667	2881	6.458
6	80	176	3200	3456	7.638
7	93	206	3733	4032	80.833
8	106	235	4266	4607	6.597
9	120	265	4800	5185	94.5141
All	133	294	5333	5760	11.181

s	$i_{s,1}$	$i_{s,2}$	$i_{s,3}$	$i_{s,4}$	$i_{s,5}$	$i_{s,6}$	Pattern neurons	E_{test} [%]
1	9	6	5	4	4	2	30	26.388
2	18	11	10	8	8	3	58	15.277
3	27	17	14	11	11	5	87	18.055
4	36	23	19	15	15	6	114	11.111
5	45	29	24	19	19	8	142	12.500
6	53	34	29	23	23	10	172	11.111
7	62	40	34	27	27	11	201	8.333
8	71	46	38	30	30	13	229	13.888
9	80	51	43	34	34	14	258	18.055
All	89	57	48	38	38	16	286	13.888

Table 4. The number of cluster centres used in determining PNN structure and the test error for DWBC data set (left table) and OC data set (right table)

s	$i_{s,1}$	$i_{s,2}$	Pattern neurons	E_{test} [%]
1	17	29	46	23.009
2	34	57	91	25.664
3	51	86	137	9.735
4	68	114	182	43.363
5	85	143	228	30.973
6	102	172	274	50.442
7	119	200	319	35.398
8	136	229	365	49.558
9	153	257	410	21.239
All	170	286	456	32.743

s	$i_{s,1}$	$i_{s,2}$	Pattern neurons	E_{test} [%]
1	11	5	16	60.000
2	21	11	32	25.000
3	32	16	48	32.500
4	42	22	64	17.500
5	53	27	80	7.500
6	63	32	95	30.000
7	74	38	112	20.000
8	84	43	127	12.500
9	95	49	144	20.000
All	105	54	159	15.000

4.2 Experimental Results after the Use of Algorithm 2

The second approach of PNN reduction consists in extracting the set of support vectors (SVs) out of the entire original data set and setting SVs as the network's pattern neurons. The process of SVs selection is performed according to **Algorithm 2**. The verification of C and sc settings requires a vast number of experiments.

The grid search for both C constraint and sc spread constant is performed: $A_C = \{10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4, 10^5, 10^6\}$ and $A_{sc} = \{0.08, 0.2, 0.3, 0.5, 0.8, 1.2, 1.5, 2, 5, 10, 50, 80, 100, 200, 500\}$. The optimal values of (C^*,sc^*) are computed as follows:

$$(C^*, sc^*) = \arg \min_{(C,sc) \in A_C \times A_{sc}} \{E_{test}\,(C, sc)\} \tag{10}$$

where E_{test} is the test error and C^*, sc^* are computed numerically.

The results are shown in Tables 5–6. From these tables one can read the number of support vectors used to construct the PNN's pattern layer and the lowest test errors calculated by the modified network. Two bottom rows indicate the test error for the original PNN and best results obtained by means of **Algorithm 1**.

One can observe that the use of the support vectors as the pattern neurons provides the decrease in the test error of PNN in all data classification cases.

Table 5. The number of support vectors used in determining PNN structure and the test error for WBC, PID, HS and CTG data sets

	WBC		PID		HS		CTG	
sc	$C^* = 10^4$		$C^* = 10^2$		$C^* = 10^0$		$C^* = 10^3$	
	SVs	E_{test} [%]	SVs	E_{test} [%]	SVs	E_{test} [%]	SVs	E_{test} [%]
0.08	43	13.138	319	64.935	135	26.229	162	86.117
0.2	48	15.328	314	64.935	134	26.229	144	48.705
0.3	47	14.598	310	64.935	136	26.229	142	26.588
0.5	47	10.218	304	64.935	133	26.229	152	11.294
0.8	48	18.978	300	64.935	137	26.229	172	25.176
1.2	49	13.868	303	64.935	138	26.229	207	32.705
1.5	54	10.948	305	64.935	134	26.229	240	18.823
2	71	14.598	302	44.155	135	26.229	288	7.058
5	182	9.489	300	48.051	139	26.229	557	24.47
10	244	8.029	333	36.363	142	26.229	909	22.352
50	329	10.218	551	31.168	171	27.868	1601	14.823
80	357	10.218	593	31.818	181	36.065	1662	15.058
100	361	9.489	607	31.818	183	32.786	1673	17.176
200	368	9.489	614	31.818	200	31.147	1682	17.176
500	368	9.489	614	31.818	220	32.786	1690	15.529
All	546	9.489	614	31.818	245	31.147	1701	15.529
Best k-means	55	6.569	368	28.571	25	24.590	1021	11.529

Table 6. The number of support vectors used in determining PNN structure and the test error for T, D, DWBC and OC data sets

	T		D		DWBC		OC	
sc	$C^* = 10^1$		$C^* = 10^3$		$C^* = 10^6$		$C^* = 10^{-1}$	
	SVs	E_{test} [%]	*SVs*	E_{test} [%]	SVs	E_{test} [%]	*SVs*	E_{test} [%]
0.08	785	40.138	86	34.722	31	21.239	112	15.000
0.2	764	15.208	110	11.111	36	62.832	113	15.000
0.3	765	11.458	133	12.500	36	7.965	117	12.500
0.5	755	16.388	158	8.333	39	27.434	119	17.500
0.8	754	8.958	186	6.944	47	19.469	121	15.000
1.2	790	8.125	214	8.333	54	10.619	130	17.500
1.5	815	9.167	239	8.333	65	23.894	137	20.000
2	874	7.500	262	9.722	78	57.522	139	17.500
5	1014	14.931	285	13.888	174	55.752	148	15.000
10	1103	6.875	286	13.888	276	49.558	153	15.000
50	1701	68.125	286	13.888	456	32.743	157	15.000
80	2045	40.763	286	13.888	456	32.743	158	15.000
100	2242	23.958	286	13.888	456	32.743	158	15.000
200	2954	71.458	286	13.888	456	32.743	158	15.000
500	4209	7.916	286	13.888	456	32.743	158	15.000
All	5760	11.181	286	13.888	456	32.743	159	15.000
Best k-means	2881	6.458	201	8.333	137	9.735	80	7.500

5 Conclusions

In this article, we considered the problem of the minimization of the number of PNN pattern layer neurons. This problem was solved along with the maximization of the generalization ability of the network. For this purpose we proposed two heuristic algorithms. The first solution relied on k-means input data clustering and the use of the cluster centres as the pattern layer neurons. In the second method, by performing SVM data classification, we determined the set of support vectors and we merely allowed the support vectors to represent the nodes in the pattern layer.

The reduced PNN models were compared with standard PNN in the classification problem of seven commonly available medical databases and one authors' own data set. In each case, the networks prediction ability was verified by computing the test error on 20% of the samples randomly separated from entire data set. The results presented in this contribution confirmed the validity of PNN structure reduction of both proposed methods. It was shown that in all data classification tasks, the reduction of the number of pattern layer neurons improved the prediction ability of the network.

It is highly probable to obtain better results, i.e.: both, smaller number of pattern neurons and the lower generalization error, after shrinking the grid search.

Because of the limited space of the article, the results are only shown for E_{test}. Similar study was performed for additional performance measures: the sensitivity, the specificity and the area under the receiver operating characteristic. The values of these measures were also better for reduced PNN.

Acknowledgements. This work was supported in part by the National Science Centre (Poland) under Grant No. NN 514 705540.

References

1. Specht, D.F.: Probabilistic Neural Networks. Neural Networks 3, 109–118 (1990)
2. Specht, D.F.: Enhancements to the probabilistic neural networks. In: IEEE International Joint Conference on Neural Networks, pp. 761–768. IEEE Press, Baltimore (1992)
3. Chtioui, Y., Bertrand, D., Barba, D.: Reduction of the size of the learning data in a probabilistic neural network by hierarchical clustering. Application to the discrimination of seeds by artificial vision. Chemometrics and Intelligent Laboratory Systems 35, 175–186 (1996)
4. Franti, P., Kaukoranta, T., Shen, D.-F., Chang, K.-S.: Fast and Memory Efficient Implementation of the Exact PNN. IEEE Trans. Image Processing 9, 773–777 (2000)
5. Adeli, H., Panakkat, A.: A probabilistic neural network for earthquake magnitude prediction. Neural Networks 22, 1018–1024 (2009)
6. Gorunescu, F., Gorunescu, M., El-Darzi, E., Gorunescu, S.: An evolutionary computational approach to probabilistic neural network with application to hepatic cancer diagnosis. In: IEEE Symposium on Computer-Based Medical Systems, pp. 461–466. IEEE Press, Dublin (2005)
7. Shan, Y., Zhao, R., Xu, G., Liebich, H.M., Zhang, Y.: Application of probabilistic neural network in the clinical diagnosis of cancers based on clinical chemistry data. Analytica Chimica Acta 471, 77–86 (2002)
8. Huang, C.-J., Liao, W.-C.: A Comparative Study of Feature Selection Methods for Probabilistic Neural Networks in Cancer Classification. In: IEEE International Conference on Tools with Artificial Intelligence, pp. 451–458. IEEE Press, Sacramento (2003)
9. Mantzaris, D., Anastassopoulos, G., Adamopoulos, A.: Genetic algorithm pruning of probabilistic neural networks in medical disease estimation. Neural Networks 24, 831–835 (2011)
10. Rutkowski, L.: Adaptive Probabilistic Neural Networks for Pattern Classification in Time-Varying Environment. IEEE Trans. Neural Networks 15, 811–827 (2004)
11. Traven, H.G.C.: A Neural Network Approach to Statistical Pattern Classification by 'Semiparametic' Estimation of Probability Density Functions. IEEE Trans. Neural Networks 2, 366–377 (1991)
12. Burrascano, P.: Learning Vector Quantization for the Probabilistic Neural Network. IEEE Trans. Neural Networks 2, 458–461 (1991)
13. Streit, R.L., Luginbuhl, T.E.: Maximum Likelihood Training of Probabilistic Neural Networks. IEEE Trans. Neural Networks 5, 764–783 (1994)
14. Zaknich, A.: A vector quantisation reduction method for the probabilistic neural network. In: IEEE International Conference on Neural Networks, pp. 1117–1120. IEEE Press, Huston (1997)

15. Berthold, M.R., Diamond, J.: Constructive training of probabilistic neural networks. Neurocomputing 19, 167–183 (1998)
16. Mao, K.Z., Tan, K.-C., Ser, W.: Probabilistic Neural-Network Structure Determination for Pattern Classification. IEEE Trans. Neural Networks 11, 1009–1016 (2000)
17. Specht, D.F., Romsdahl, H.: Experience with adaptive probabilistic neural networks and adaptive general regression neural networks. In: IEEE World Congress on Computational Intelligence 2, pp. 1203–1208. IEEE Press, Orlando (1994)
18. Parzen, E.: On estimation of a probability density function and mode. Annals of Mathematical Statistics 36, 1065–1076 (1962)
19. Masters, T.: Practical Neural Networks Recipes in C++. Academic Press, San Diego (1993)
20. Hartigan, J.A., Wong, M.A.: A k-means clustering algorithm. J. Royal Stat. Soci.–Ser. C (Applied Statistics) 1, 100–108 (1979)
21. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.-H., Steinbach, M., Hand, D.J., Steinberg, D.: Top 10 algorithms in data mining. Knowledge Information Systems 14, 1–37 (2008)
22. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
23. Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Neural Computation 12, 1207–1245 (2000)
24. Gunn, S.R.: Support Vector Machines for Classification and Regression. Technical Report, University of Southampton (1998)
25. Schölkopf, B., Burges, C.J.C., Smola, A.J.: Advances in Kernel Methods – Support Vector Learning. MIT Press, Cambridge (1999)
26. Scholköpf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2002)
27. UCI Machine Learning Repository, `archive.ics.uci.edu/ml/datasets.html`
28. Skret, A., Lozinski, T., Chrusciel, A.: Epidemiology of ovarian cancer: Hormonal and genetic aspects. In: Congress of the European Society for Gynecologic and Obstetric Investigation, pp. 189–205. CIC Edizioni Internazionali, Rome (2001)