Combining Complementary Neural Network and Error-Correcting Output Codes for Multiclass Classification Problems

JAIRAJ PROMRAK
Mahidol University
Department of Mathematics
Faculty of Science
Bangkok, THAILAND
jpjairaj@gmail.com

PAWALAI KRAIPEERAPUN
Ramkhamhaeng University
Department of Computer Science
Faculty of Science
Bangkok, THAILAND
pawalai@yahoo.com

SOMKID AMORNSAMANKUL
Mahidol University
Department of Mathematics
Faculty of Science
Bangkok, THAILAND
and Centre of Excellence in Mathematics
CHE, Sriayudhaya Rd., Bangkok, Thailand
scsam@mahidol.ac.th

Abstract: This paper presented an innovative method, combining Complementary Neural Networks (CMTNN) and Error-Correcting Output Codes (ECOC), to solve multiclass classification problem. CMTNN consist of truth neural network and falsity neural network created based on truth and falsity information, respectively. In the experiment, we deal with feed-forward backpropagation neural networks, trained using 10 fold cross-validation method and classified based on minimum distance. The proposed approach has been tested with three benchmark problems: balance, vehicle and nursery from the UCI machine learning repository. We found that our approach provides better performance compared to the existing techniques considering on either CMTNN or ECOC.

Key–Words: Multicass Classification, Complementary Neural Network, Error-Correcting Output Codes (ECOC), Feed-forward Backpropagation Neural Network

1 Introduction

Classification is one of the most frequently used in the field of artificial intelligence [17]. A classification problem occurs when we need to assign an object into a group or class based on its observed properties. The task of classifying objects into two groups is known as binary classification whereas the assignment of an object into one of several classes (more than 2 classes) is called multiclass classification. Classification problems appear in several areas such as science, medicine, industry, and business. In order to solve this kind of problem, one of the most successful and popular methods is Neural Networks (NN) [1].

Applications of neural network to classification have been widely studied. For example, neural networks can be considered as data driven self-adaptive methods, universal functional approximators, and posterior probabilities estimator [2, 7, 17]. The effectiveness of neural network classification has been tested empirically. Neural networks have been successfully applied to a variety of real world classification problems [15]. In 2008-2010, there are several academic publications such as in branches of fault detection, medical diagnosis, and bankruptcy prediction [16, 12, 5]. Although several types of neural networks can be used for classification purposes [13], our focus is on the feed-forward neural network since it has been widely studied and used to solve the classification problem. In order to solve multiclass classification problems, feed-forward neural network can be integrated with other techniques such as codeword designs and decoding [6].

In designing code, to avoid non-unique extension of binary classification to multiclass classification, four approaches have been suggested: one-against-all, pairwise comparison, all-at-once classifications, and Error-Correcting Output Code (ECOC) [8]. Dietterich and Bakiri [3] found that applying ECOC to multiclass learning problems can improve generalization capabilities of classification systems. Moreover, it was found that ECOC can reduce bias and variance which occur when using decision-tree learning algorithm [9].

In recent years, Complementary Neural Networks (CMTNN) based on feedforward neural network have been proposed to deal with both binary and multiclass classification problems [10, 11]. Instead of considering only the truth information, CMTNN considered both truth and falsity information in order to enhance the classification results. CMTNN consist of truth neural network and falsity neural network created based on truth and falsity information, respectively. It was found that CMTNN provided better performance when compared to the traditional feed-

ISBN: 978-960-474-281-3

forward neural networks [10, 11].

In this paper, we aim to integrate ECOC with CMTNN to solve multiclass classification problems. They can be described as follows.

1.1 ECOC for Multiclass Learning Problems

ECOC is an information theoretic concept used to correct errors when transmitting data in communication tasks. The idea of these codes is adding some redundant cases which do not match with any acceptable solution in the output set. If one of these cases appears after data is transmitted, the system will realize occurence of the error. Simple set rules used to form the code depending on the number of class (k) have been presented which are

• exhaustive codes for $3 \leq k \leq 7$,

• column selection from exhaustive code for $8 \leq k \leq 11$, and

• randomized hill climbing and Bose-Chaudhuri-Hocquengham (BCH) codes when $k > 11$ [4].

In this work, we focus only on the exhaustive code described in [4]. It can be explained as follows.

For data set having k classes when $3 \leq k \leq 7$, a code of length $2^{k-1} - 1$ can be constructed where

class 1 : All strings are one,

class 2 : There are 2^{k-2} zeroes followed by $2^{k-2} - 1$ ones,

class 3 : There are 2^{k-3} zeroes, followed by 2^{k-3} ones, followed by $2^{k-3} - 1$ zeroed, followed by $2^{k-3} - 1$ ones.

$\vdots \qquad \vdots$

class i : There are alternating runs of 2^{k-i} zeroes and ones.

For example, when $k = 4$; length of code has $2^{4-1} - 1 = 7$ digits where class 1 is 1111111, class 2 has $2^{4-2} = 4$ zeroes and $2^{4-2} - 1 = 3$ ones, class 3 has $2^{4-3} = 2$ zeroes, $2^{4-3} = 2$ ones, $2^{4-3} = 2$ zeroes and $2^{4-3} - 1 = 1$ one, and class 4 has $2^{4-4} = 1$ zeroes, $2^{4-4} = 1$ ones, $2^{4-4} = 1$ zeroes, $2^{4-4} = 1$ one, $2^{4-4} = 1$ zeroes, $2^{4-4} = 1$ ones, and $2^{4-4} = 1$ zeroes; see table 1.

Class	**Codeword**
1	1 1 1 1 1 1 1
2	0 0 0 0 1 1 1
3	0 0 1 1 0 0 1
4	0 1 0 1 0 1 0

Table 1: Exhaustive codes when $k = 4$

In our experiment, we compare results obtaining from exhaustive code to the One-Per-Class (OPC) code. OPC code can be described as follows.

For data set having k classes, the OPC code of class i is the codeword with k digits where i^{th} digit is one and other digits are zero for $i = 1, 2, 3, ..., k$. That is,

$$
\begin{aligned}
&\text{class 1 has the codeword } \overbrace{1\ 0\ 0\ 0\ 0 \cdots\ 0}^{k \text{ digits}},\\
&\text{class 2 has the codeword } 0\ 1\ 0\ 0\ 0 \cdots\ 0,\\
&\text{class 3 has the codeword } 0\ 0\ 1\ 0\ 0 \cdots\ 0,\\
&\qquad \vdots\\
&\text{class } i \text{ has the codeword } 0 \cdots\ 0 \underbrace{\ 1\ }_{i^{th} \text{ digit}} 0 \cdots\ 0.
\end{aligned}
$$

The examples of OPC code's designs for $k = 4$ are shown in table 2.

Class	**Codeword**
1	1 0 0 0
2	0 1 0 0
3	0 0 1 0
4	0 0 0 1

Table 2: One per class codes when $k = 4$

1.2 Complementary Neural Networks

In order to solve the classification problem using neural network, on the whole, we deal with binary values, 0 and 1, representing the truth information. However, it is not exactly true. Hence, degrees of truth must be considered. For degree of truth, its value is in the set $[0, 1]$. Instead of considering only the truth information, the complement of the truth which is the falsity information should also be considered since the predicted output may not exactly true. Therefore, the truth neural network and the falsity neural network are created in order to predict the truth output and the falsity output, respectively. These two output are predicted in the sense that they should be complement to each other. If both output values are similar then it is an indicator that we may have to readjust parameters of neural networks. The combination of these two networks is called complementary neural networks (CMTNN). Both truth and falsity neural networks are created based on the implication rules shown in table 3.

Type of NN	**Input**	**Target**	**Output (Inference)**
Truth NN	True	True	True
Falsity NN	True	False	False

Table 3: Implication rules

From the table, the logical implication "if X then Y ($X \rightarrow Y$)" is applied. If we know that X and Y are true, we then get its inference also true. On the other hand, if X is true but Y is false, then its inference is false. In the training phase of CMTNN, X and Y are considered as the input feature and the target value, respectively. The inference is considered as the predicted output.

Suppose that we have n patterns, each with m features, and we want to classify patterns into k classes. For each pattern, let x_i be the input pattern where $i = 1, 2, 3, ..., n$, $t(x_i)$ be the truth target value, $f(x_i)$ be the falsity target value, $T(x_i)$ be the truth output value, $F(x_i)$ be the falsity output value. The process of solving multiclass classification using complementary neural networks is shown in figure 1.

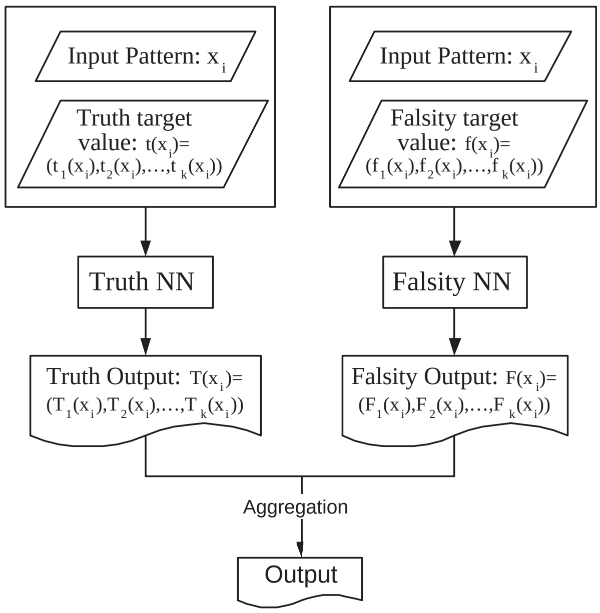

Figure 1: Complementary neural networks model

After the input patterns, the truth and falsity target values are entered, the truth and falsity neural networks are trained to predict degrees of truth and falsity output value separately. After that, both trained networks can be used to predict the unknown input pattern. The truth and falsity output obtained from both trained networks can be aggregated to form the final output which is then classified into one of the k classes.

2 Proposed Methodologies for Multiclass Classification

2.1 Data Sets

In the experiments, data sets used to test the proposed multiclass classification model are summarized in the table 4. All data sets are obtained from UCI machine learning repository which is publicly available [14]. They all are of type classification having number of classes from three to seven with none of any missing values. Each raw data set is normalized to numeric form in the set of $[0, 1]$. Then, data set will be separated into two parts for training phase and testing phase using 10-fold average method.

Name	Balance	Vehicle	Nursery
# classes	3	4	5
# features	4	18	8
# samples	625	846	12,960

Table 4: Characteristics of the selected data sets from the UCI repository

2.2 Classification Model

The classification algorithm is separated into two parts: training phase and testing phase in which each phase can be explained as follows.

Let x be a vector of m features, C_t be a set of codewords for truth information, C_f be a set of codewords for falsity information, D_t be a truth training set and D_f be a falsity training set which are denoted by $x = (x_1, x_2, x_3, ..., x_m) \in X$,
$C_t = \{c_{t_1}, c_{t_2}, c_{t_3}, ..., c_{t_k}\}$,
$C_f = \{c_{f_1}, c_{f_2}, c_{f_3}, ..., c_{f_k}\}$,
$D_t = \{< x, c_t >\in X \times C_t\}$ and
$D_f = \{< x, c_f >\in X \times C_f\}$, respectively.
In training phase, learning algorithm is used by input D_t, constituting of feature vectors and its truth target codeword vectors, to learn a classifier function $L_t : X \rightarrow C_t$. In the same manner to falsity information, the function $L_f : X \rightarrow C_f$ is obtained using the input set D_f. The process of training phase of CMTNN model is shown in figure 2.

CMTNN consists of truth neural network (Truth NN) and falsity neural network (Falsity NN) where architectures and properties of Truth NN and Falsity NN are the same. However, the truth NN is trained from target codeword vectors to predict truth output but falsity NN is trained from complementary of target codeword vectors to predict falsity output. In this paper, we focus only on the exhaustive code where $3 \leq k \leq 7$. For example, consider OPC code with $k = 3$.

ISBN: 978-960-474-281-3

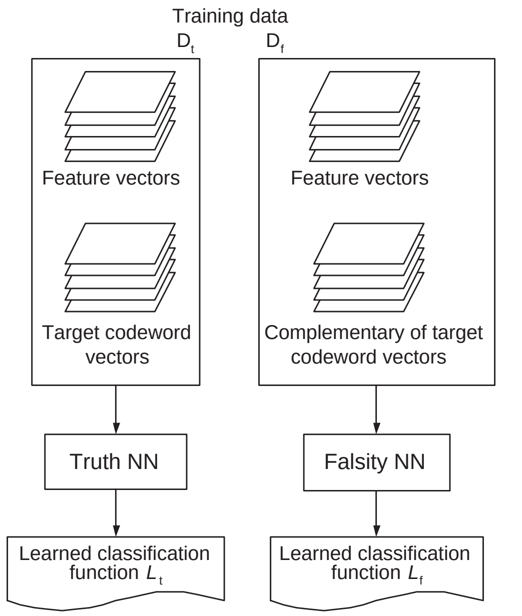

Figure 2: Training phase for complementary neural networks model

The target codeword vector is $C_t = (100, 010, 001)$ and its complement is $C_f = (011, 101, 110)$.

In testing phase, the rest data set will be used. Features of a new sample are transferred through the truth NN and falsity NN basing on the learned classification function from truth NN and falsity NN, respectively. Then the truth and falsity outputs, $T(x)$ and $F(x)$, are obtained. The testing phase algorithm for CMTNN is shown in figure 3.

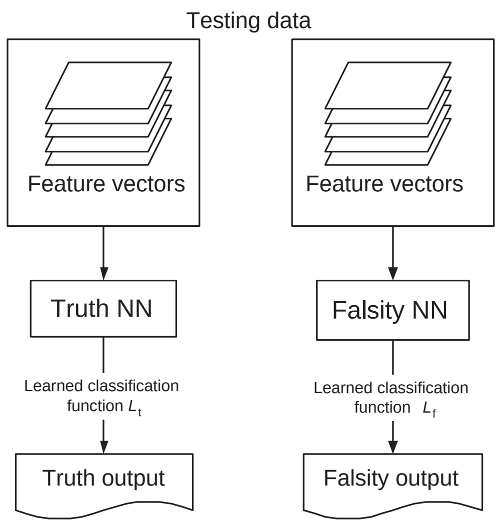

Figure 3: Testing phase for complementary neural networks model

Since the predicted codeword $P(x)$, aggregating from the truth and falsity output, may not perfectly match to the predefined class, the decoding method is required for mapping $P(x) \rightarrow C_t$.

2.3 Decoding

Let $c_t = c_t^1 c_t^2 c_t^3 ... c_t^j$, $c_f = c_f^1 c_f^2 c_f^3 ... c_f^j$, $T_1 T_2 T_3 ... T_j \in T(x)$ and $F_1 F_2 F_3 ... F_j \in F(x)$ where j is length of the codeword. The decoding technique basing on minimum distance is applied and described below.

For fixed x, find the distance d between output and each predefined class for CMTNN model as following:

$$d = \sum_{i=1}^{j} |T_i - c_t^i| + \sum_{i=1}^{j} |F_i - c_f^i|.$$

Note that if we decode on the neural network dealing only with the truth output, there are only one term in the right hand side. Then the class with minimum distance is chosen.

After the predicted class is obtained from decoding, it will be compared to the truth target codeword to evaluate an efficiency of the proposed method which is shown in the next topic.

3 Experimental Methodology and Results

In the experiment, 10 fold cross-validation method is applied. In each neural network, $2m$ neurons are created in the hidden layer. The networks are trained using backpropagation where Mean Squared Error (MSE) is considered as the objective functions. Feed-forward backpropagation method is used for training all the networks 5,000 iterations with a learning rate of 0.5.

The accurate percentages of classification from each technique is compared and shown in table 5. This table is separated into two parts: the first part shows the results obtained from NN model using OPC code and exhaustive code. We found that all data sets give better results when applying exhaustive code. The second part of table 5 shows the results obtained from CMTNN in which two types of codeword are also compared. We found the same consequence to NN that the exhaustive code provides better performance when compared to the OPC code. From both parts, we found that CMTNN gives better results when using OPC code as a codeword. In addition, in the exhaustive code, CMTNN model improves the classification performance as compared to NN model for all

ISBN: 978-960-474-281-3

Technique		Balance	Vehicle	Nursery
architecture	codeword			
NN	OPC code	83.72	56.59	87.95
	exhaustive code	89.45	60.17	89.91
CMTNN	OPC code	91.68	68.09	88.04
	exhaustive code	92.00	82.62	91.00

Table 5: The percentage of average classification accuracy for the test set obtained by applying different techniques.

Technique	Balance	Vehicle	Nursery
NN with OPC code	9.89	56.60	3.47
NN with exhaustive code	2.85	37.31	1.21
CMTNN with OPC code	0.35	21.34	3.36

Table 6: The percentage of increased average classification accuracy for the test set comparing to the purposed technique (combining CMTNN with exhaustive code)

data set. Table 6 shows the increased percent accuracy of each method comparing to our proposed technique. Therefore, we can conclude empirically that the CMTNN model provides better performance than NN model and ECOC technique gives better results than OPC code. Moreover, as we desire, the results obtained from the proposed technique, combination between CMTNN model and ECOC method, outperforms the existing techniques shown in table 5.

4 Conclusion

This paper has applied the CMTNN and exhaustive code to solve k-class classification problems for $3 \leq k \leq 7$. The proposed method are tested using three data sets from UCI machine learning repository database. The results show that our method provides better accuracy percentage than traditional techniques based on only CMTNN or ECOC where using the minimum distance as a decoding. In the future, we will apply our method to multiclass classification problem with $k > 7$ and consider other decoding techniques as well.

Acknowledgements: The research was supported by Faculty of Science, Mahidol University and in the case of the first author, it was also supported by the Development and Promotion of Science and Technology talents project (DPST).

References:

[1] Christopher M. Bishop. *Neural Network for Pattern Recognition.* Oxford, 2005.

[2] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Contr. Signals Syst.*, 2:303–314, 1989.

[3] T. Dietterich and G. Bakiri. Error - correcting output codes: A general method for improving multiclass inductive learning programs. In *Proceedings of AAAI-91*, pages 572–577. AAAI Press / MIT Press, 1991.

[4] T.G. Dietterich and G. Bakiri. Solving multiclass learning problem via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[5] H. Fu-yuan. A genetic fuzzy neural network for bankruptcy prediction in chinese corporations. *Proc. of the 2008 International Conference on Risk Management & Engineering Management*, pages 542–546, 2008.

[6] S. Haykin. *A Comprehensive Foundation*, chapter Neural Networks. Macmillan, New York, 1994.

[7] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991.

[8] T. Kikuchi and S. Abe. Comparison between error correcting output codes and fuzzy support vector machines. *Pattern Recognition Letters*, 26:1937–1945, 2005.

[9] E. Kong and T. G. Dietterich. Error-correcting output coding correct bias and variance. In *The*

XII International Conference on Machine Learning, pages 313–321, San Francisco, 1995.

[10] P. Kraipeerapun and C.C. Fung. Binary classification using ensemble neural networks and interval neutrosophic sets. *Neurocomputing*, 72:2845–2856, 2009.

[11] P. Kraipeerapun, C.C. Fung, and K.W. Wong. Uncertainty assessment using neural networks and interval neutrosophic sets for multiclass classification problems. *WSEAS Transactions on Computers*, 6:463–470, 2007.

[12] J. Lan, M.Y. Hu, E. Patuwo, and G.P. Zhang. An investigation of neural network classifiers with unequal misclassification costs and group sizes. *Decision Support Systems*, 48:582–591, 2010.

[13] R.P. Lippmann. Pattern classification using neural networks. *IEEE Commun. Mag.*, 27:47–64, 1989.

[14] D. Newman, S. Hettich, C. Blake, and C. Merz. Uci repository of machine learning databases. online, November 2009. http://archive.ics.uci.edu/ml/datasets.html.

[15] B. Widrow, D.E. Rumelhard, and M.A. Lehr. Neural networks: Applications in industry, business and science. *Commun. ACM*, 37:93–105, 1994.

[16] J. Wu and C. Liu. An expert system for fault diagnosis in internal combustion engines using wavelet packet transform and neural network. *Expert Systems with Applications*, 36:4278–4286, 2009.

[17] G.P. Zhang. Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics*, 30:451–462, 2000.