MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

A Pattern Classifier Integrating Multilayer Perceptron and Error-Correcting Code

Haibo Li, Torbjörn Kronander, and Ingemar Ingemarsson
Department of Electrical Engineering,
Linköping University, S-58183 Linköping, Sweden
e-mail: Haibo@isy.liu.se, tobbe@isy.liu.se

Abstract

In this paper we present a novel classifier which integrates a multilayer perceptron and a error-correcting decoder. There are two stages in the classifier, in the first stage, mapping feature vectors from feature space to code space is achieved by a multilayer perceptron; in the second stage, error correcting decoding is done on code space, by which the index of the noisy codeword can be obtained. Hence we can get classifications of original feature vectors. The classifier has better classification performances than the conventional multilayer perceptrons.

1 Introduction

Multilayer perceptrons trained with backpropagation[4] have been successfully applied in many areas[4][5][6][7], especially in pattern recognition[5]. A number of theoretical analyses have shown that any continues nonlinear mapping can be closely approximated using sigmoid nonlinearities and multilayer perceptron that implies that arbitrary decision regions can be formed by multilayer perceptrons. For some real-world problems, however, it is very difficult to train a multilayer perceptron for forming mapping needed within given accurate, especially when the decision regions required are more complex and irregular, even if neural networks have more hidden layers and enough time be provided to train it.

Meanwhile Error-Correcting Code theory has been well established a long time. The error correcting decoding can be viewed as a kind of classifying problem in which the noisy codewords are decoded into the correct codewords based on minimum Hamming distance in code space. With minimum Hamming distance criterion, the decision regions in code space are very regular, for example are circle. The mapping from feature space to code space is easier achieved because such mapping is from region to region instead of from region to point. The classifier has better classification performancs than the conventional multilayer perceptrons.

In this paper, we focus on how to form mapping from feature space to code space with a multilayer perceptron and how to seek good error correcting codes whose codeword distributions are uniform and codeword distance is large.

We will below first describe the pattern Classifier architecture. Then we discuss how to train a multilayer perceptron and outline the error-correcting code. We conclude with remarks on our pattern Classifier and future work.

2 Classifier Architecture

Before we describe our new classifier, let's first examine the pattern classifier problem.

For an N-dimensional feature space B^N, assuming there are M categories pattern in this space, this is, this feature space can be partitioned into M cells $C_i, i = 1, 2, ..., M$ and associates to each cell C_i a exemplar vector $\vec{f_i}$.

For a given feature vector $\vec{X} \in B^N$, we say it belongs to ith category C_i, if it meets following condition

$$|\vec{X} - \vec{f_i}| < |\vec{X} - \vec{f_i}| \quad \forall i \neq j \tag{1}$$

where $|.|$ denotes some kind of distance measure in B^N, Hamming distance is used here.

This pattern classification problem can be solved by a multilayer perceptron with the backpropagation algorithm as its learning algorithm[5]. The input of the multilayer perceptron is $\vec{X} \in B^N$, the output is $\vec{Y}$. Supposing the desired exemplar is $\vec{d}$, then the error function can be defined as following

$$E = |\vec{d} - \vec{Y}| \tag{2}$$

then the weights of the multilayer perceptron are changed in the sequence by an amount proportional to the partial derivatives of E with respect to the weights until a error principle is met. The general error principle is

$$E < \epsilon \tag{3}$$

where ϵ is a fault tolerant threshold.

For our pattern classification problem, $\vec{d}$ and $\vec{Y}$ are binary vector, so the ϵ used here is less than the minimum Hamming distance[1][2]. The desired exemplar vector $\vec{d}$ is generally given with the minimum Hamming distance 1. If the minimum Hamming distance is great than 1, the classification results will be in confusion provided that any postprocessing is not used for the results. The minimum Hamming distance is less than 1 which means $\epsilon < 1$, a pattern classification with this method can be viewed as a kind mapping from a region to a point. This may be the reason why the multilayer perceptrons are especially difficult to train for some real-world problems.

Based on the above analysis, we present a novel pattern classifier which integrates a multilayer perceptron and a error-correcting decode. The idea behind the classifier is achieving classification through, first, region space mapping, that is, mapping from a region of feature space to a region of code space by a multilayer perceptron, then error-correcting decoding is done in the code space, we can obtain the index of classification. The main difference between the new classifier and multilayer perceptron is mapping from region to region instead of mapping from region to point in our new classifier.

For a pattern classification problem with M categories, we can find a set error-correcting code with M codewords, $D_i, i = 1, 2, ..., M$, whose word length are L when the minimum Hamming distance H_{min} is predetermined. The wordlength L is determined by M and H_{min}.

The desired signals used for training classifier are codewords $D_i, i = 1, 2, ..., M$, the error function is defined as follows

$$E = |D - \vec{Y}| \tag{4}$$

where $\vec{Y}$ is the output of the multilayer perceptron.

The error principle used here is also described by (3), but we can find that the minimum Hamming distance is free to choose, it can be great than 1. Clearly, it is easier to train the multilayer perceptron than the one without mapping in code space. Because it is easier to meet the error principle, it not only improve the accurate but speed the training rate.

After the patterns are mapped onto code space, we can make error-correcting decoding in the code space, we can obtain the index of patterns. The process of a pattern classification by the classifier is shown in Fig.1.

The overall classifier structure consists of two levels: the first level maps pattern vectors from feature space onto code space by a multilayer perceptron; the second level is a error

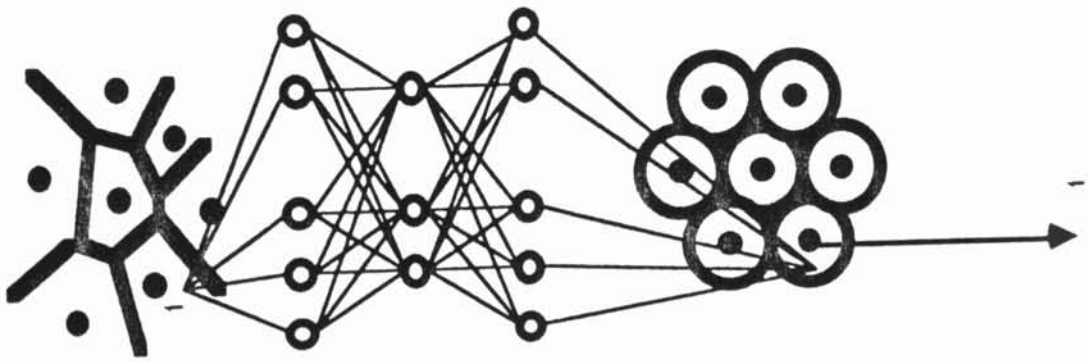

Figure 1: The process of a pattern classification by the new classifier

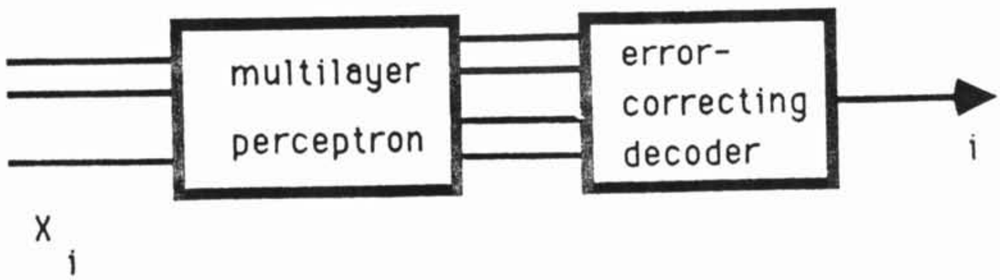

Figure 2: The structure of the new classifier

correcting decoder which indicates the index of the input pattern. Such a system is shown in Fig.2.

3 Mapping from feature space to code space

This section will mainly cover how to map a feature vector from feature space to code space by a multilayer perceptron.

Tzi-Dar Chiueh and R.Goodman[3] gave two schemes, outer product method and pseudo-inverse method, mapping from feature space to code space. These methods are simple to implement but strictly speaking in theory they can't achieve strict mapping, i.e.

For a given feature vector $\vec{X} \in C_i$, that is

$$|\vec{X} - \vec{f_i}| < |\vec{X} - \vec{f_j}| \quad \forall i \neq j \tag{5}$$

Now $\vec{X}$ is mapped onto code space,

$$\vec{Y} = W\vec{X} \tag{6}$$

where W is a transformation matrix satisfying the following equation,

$$\vec{D_i} = W\vec{f_i} \quad \forall i \tag{7}$$

but it is difficult to ensure the following equation valid for any $\vec{X}$

$$|\vec{Y} - \vec{D_i}| < |\vec{Y} - \vec{D_j}| \quad \forall i \neq j \tag{8}$$

We achieve such mapping from feature space to code space by means of a multilayer perceptron. The input of a multilayer perceptron are $\vec{X_i}$, the desired signals are $\vec{D_i}$. Then the multilayer perceptron is trained by the backpropagation algorithm. When the maximum train error is less than the minimum Hamming distance, the training of the multilayer perceptron finished.

If the exemplars, $\vec{f_i}$ of pattern vectors are known, a better train procedure be designed as follows

1) the multilayer perceptron is first trained with input $\vec{f_i}$

2) after the network is stable, the input $\vec{X}$ are chosen in line according to the shortest distance to their corresponding exemplars step by step for training the multilayer perceptron.

This kind training procedure similar to the cooling procedure used in the simulated annealing can avoid sinking in local minimum pitfall. Hence a better mapping performances can be obtained.

4 Error-correcting code

Error-correcting code has been found in a wide variety of applications, including data transmission over a communication channel, codes for high-speed and mass memories of computer ect.

The error-correcting code we need should meet the following properties 1) the distance between each two codewords must be the same; 2) the distance between each two codewords must be as large as possible.

Such codes as Hadamard matrix codes can be found in [2], due to the limitations of space, we don't discuss this content here.

5 Conclusion remarks

Integrating error-correcting code, the multilayer perceptron can give a better performance than without such codes. Of course, the improvement in the performance is at the cost of the complex system structure. The more the error we want tolerant, the larger the codewords are, and the more complex the system is. How to choose the trade-off between the complex and fault tolerance is difference from case to case. But we can say error-correcting code has a wide application foreground in neural networks. We are in the beginning of applying error-correcting codes for neural networks, we will work more with such object, since integrating neural networks and error-correcting codes seems to promise a good path in this aspect.

References

[1] R.Blahut, Theory and Practice of Error Control Codes, 1983

[2] F.Macwilliams and N.Sloane, The Theory of Error-correcting Codes, 1988

[3] T.Chiueh and R.Goodman, A Neural Network Classifier Based on Coding Theory, AIP 1988

[4] D.E.Rumelhart ect, Learning Internal Representations by Error Propagation, Parallel Distributed Processing, volume 1, MIT Press, 1986

[5] R.P.Lippmann, Pattern Classification Using Neural Networks, IEEE Comm. magazine November 1989.

[6] Haibo Li, Nonlinear Predictive Image Coding Using Neural Networks, ICASSP 1990

[7] E.Baum, On the Capabilities of Multilayer Perceptrons, J.Complexity,1988