result can be seen for higher order maps, in particular for the projection type C. Higher order SOMs' classification performance is lower than the standard SOM under no projection pursuit on account of correlated inputs i.e. higher the order of correlation lower the recognition performance.

6 Conclusions

In this paper, we have proposed a feature projection approach to enhance the classification performance of the SOMs and demonstrated on a variety of SOMs that the performance can be improved by choosing an appropriate high dimensional manifold for feature projection. Tests on real world texture classification indicate consistent performance for at least two projection types A and C, while projection type B is found to be good for SO- and TO-SOMs. Further tests to integrate these maps for improving prediction accuracies are under progress.

References

[1] V. Chandrasekaran, M. Palaniswami, and T. M. Caelli. Performance evaluation of spatio-temporal feature maps with gated neuronal architecture. In *Proceedings of the World Congress on Neural Networks*, pages IV–112–118, July 1993.

[2] V. Chandrasekaran, M. Palaniswami, and T. M. Caelli. Spatio-temporal feature maps using gated neuronal architecture. *IEEE Transactions on Neural Networks*, 6(5):1119–1131, September 1995.

[3] T. Kohonen. The self-organizing map. *Proceedings of IEEE*, 78(9):1464–1480, September 1990.

[4] G. D. Wilensky and N. Manukian. The projection neural network. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2, pages 358–367, 1992.

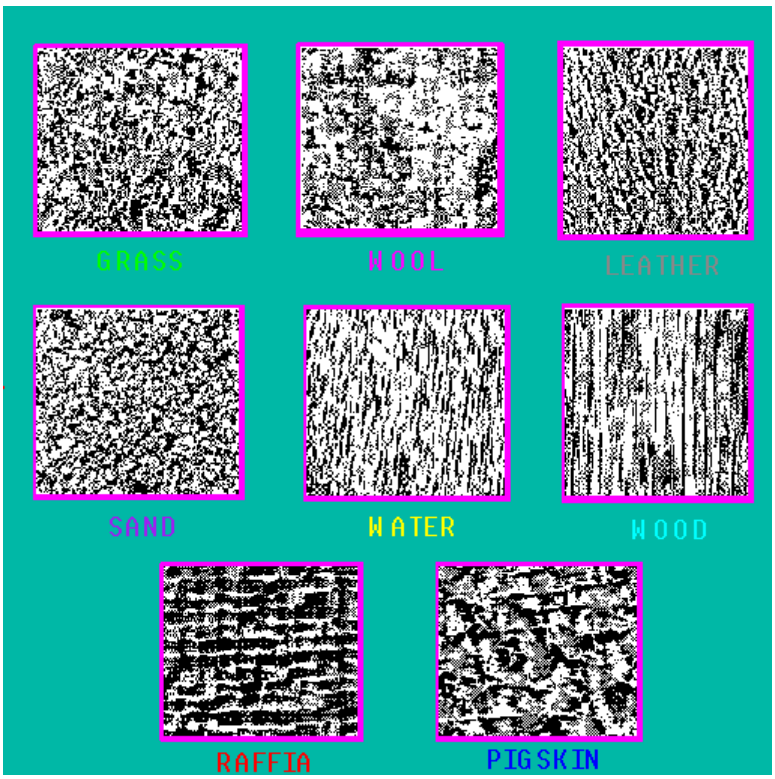

Figure 3: Texture images used in the classification experiment

4.2 SOM setup

SOMs of size 10x10 are chosen for testing the projection pursuit hypothesis. These SOMs are trained on the input feature space spanned by the 1024 vectors. The initial conditions for all the maps were neighborhood size 5, learning rate 0.05 and 50 epochs. These were modified in subsequent batch mode training as (4,0.005,100), (3,0.0005,200), (2,0.00005,400), and (1,0.000005,800). The resulting map is used for testing the classification performance.

5 Results of Simulations

The following table 1 indicates the number of patterns correctly classified for all three types of SOMs with and without projection pursuits. Percentage classification is computed over 1024 input pattern set.

TYPE	No Projection	Projection-Type A	Projection-Type B	Projection-Type C
FO-SOM	741 (72.36%)	751 (73.34%)	730 (71.29%)	747 (72.95%)
SO-SOM	696 (67.97%)	741 (72.36%)	741 (72.36%)	784 (76.56%)
TO-SOM	637 (62.21%)	748 (73.04%)	739 (72.17%)	773 (75.49%)

Table 1: Classification Performance of Different Types of SOM with Projection

In the above Table 1, FO-SOM, SO-SOM and TO-SOM indicate first, second, and third order maps. While the improvements to classification performance is marginal in first order, satisfactory

from 1 to input dimension m (extended from original dimension n). The learning rate function $\mu(\cdot)$ has values in the interval [0,1] and is a monotonically decreasing function of the discrete time index t. The winning node is based on the neuron having a least distance to the input pattern. The distance metric is given by:

$$d(\mathbf{V}, \mathbf{W}_{(x,y)}) = \sqrt{\sum_{i=1}^{m} (v_i - w_{i,(x,y)})^2} \qquad \text{for } x, y \in N_w \tag{5}$$

3.2 For second order SOM

Weight update rule for second-order SOM is given by:

$$a^{new}_{(i,j)(x,y)} = \begin{cases} a^{old}_{(i,j)(x,y)} + \beta(t)[v_i * v_j - a^{old}_{(i,j)(x,y)}] & \text{for}(x,y) \in N_w \\ a^{old}_{(i,j)(x,y)} & \text{for } (x,y) \ni N_w \end{cases} \tag{6}$$

where $a_{(i,j)(x,y)}$ represents the second order weight connection to neuron at feature map location (x, y) from ith input modulated by jth input. The learning rate $\beta(\cdot)$ for a second order map is similar to $\mu(\cdot)$ in a first order map. The distance metric in second-order map is defined to be:

$$d(\mathbf{V}, \mathbf{A}_{(x,y)}) = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{m} (v_i * v_j - a_{(i,j)(x,y)})^2} \qquad \text{for } x, y \in N_w \tag{7}$$

3.3 For third order SOM

Similar to second order SOM weight update algorithms, weight updates for third order SOFMs are detailed below for the sake of completeness.

$$b^{new}_{(i,j,k)(x,y)} = \begin{cases} b^{old}_{(i,j,k)(x,y)} + \gamma(t)[v_i * v_j * v_k - b^{old}_{(i,j,k)(x,y)}] & \text{for } (x,y) \in N_w \\ b^{old}_{(i,j,k)(x,y)} & \text{for } (x,y) \ni N_w \end{cases} \tag{8}$$

where $b_{(i,j,k)(x,y)}$ represents the third order weight connection at feature map location (x, y) from ith input location modulated by j and kth inputs. The learning rate function $\gamma(\cdot)$ is defined in a similar manner to $\mu(\cdot)$ and $\beta(\cdot)$ for first and second order SOMs respectively. The distance metric for third order map is defined to be:

$$d(\mathbf{V}, \mathbf{B}_{(x,y)}) = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} (v_i * v_j * v_k - b_{(i,j,k)(x,y)})^2} \qquad \text{for } x, y \in N_w \tag{9}$$

4 Experimental setup

4.1 Feature data set

A set of eight real world texture data is used in the classification experiment.These are obtained from the University of Southern California - Signal Processing Research Institute Database. The size of each original texture image is 512x256 pixels. For the purpose of feature extraction, each image is divided into 128 blocks of 32x32 pixels. Subimages of 16x16 were extracted from the center of each 32x32 block. Each subimage was then operated upon by 3x3 pixel masks of eight different types as per Law[?]. The operation of the mask upon each subimage created a single real value which characterized the response of the image to the mask. Hence for each subimage, a vector of eight real valued elements was obtained. The complete data set of 1024 vectors comprised eight sets of 128 feature vectors, one for each texture type. The types of texture images used in the experiment are: (i) Raffia, (ii) Grass, (iii) Pigskin, (iv) Sand, (v) Water, and (vi) Wood. The texture images are shown in Fig. 3.

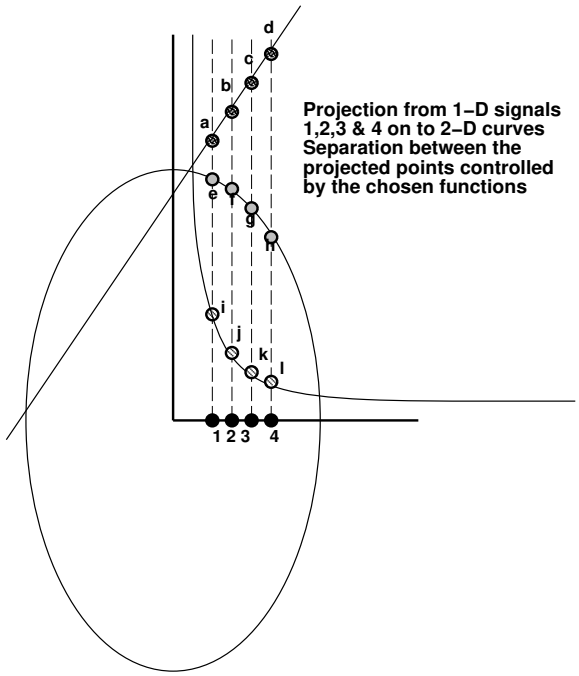

Figure 2: An example of feature space extension 1-D to 2-D

computed as below.

$$x_{n+j} = \sqrt{R^2 - x_j^2} \; \forall \; j > n \tag{2}$$

Type C: Extension to 2n dimensions:
As in type B, the extended feature vector will have 2n dimensions with each of its projected element is computed as below.

$$x_{n+j} = \sqrt{R^2 - \sum_{i=1}^{j} x_i^2} \; \forall \; j > n \tag{3}$$

The above projections will be demonstrated on three types of SOMs viz. standard Kohonen's SOM with first order connections, second order and third order SOMs. Here the connections a_{ijk} and b_{ijkl} indicate connection to neuron i with correlated inputs from j, k and l elements of the input feature vectors. For example second order neurons will have inputs of the types $x_j * x_k$ and the third order neurons will receive $x_j * x_k * x_l$ respectively. Such inputs provide modified topologies for the same input space. The definitions and the weight update rule for the above SOMs are given in the following sections.

3 Weight update rules

3.1 For Kohonen's SOM

The weight update rule is given by:

$$w_{i,(x,y)}^{new} = \begin{cases} w_{i,(x,y)}^{old} + \mu(t)[v_i - w_{i,(x,y)}^{old}] & \text{for } (x,y) \in N_w \\ w_{i,(x,y)}^{old} & \text{for } (x,y) \ni N_w \end{cases} \tag{4}$$

where v_i is the ith element of input vector V, N_w represents all neurons in the neighbourhood of the winner and (x, y) provides the coordinates of the neuron in the feature map. Index i varies

Section 2 provides the theory behind the projection pursuit and enumerates the types of projections used. Section. 3 elucidates the general weight update rule for high-order correlation maps. In section 4 details the test data, SOM setups and the learning parameters. Section 5 provides simulation results and conclusion.

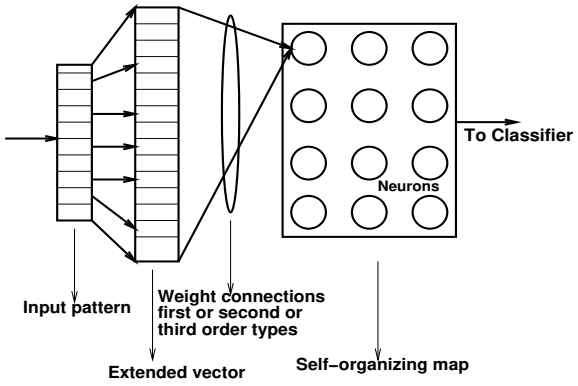

Figure 1: General Arrangement of SOM Classifier based on Projection Pursuit

2 Projection Concept

The proposed projection pursuit method solely depends on the type of feature projection onto high dimensional spheres or manifolds. The degree of separability of the features varies in accordance with the functions chosen. For example, let us consider an 1-D feature space extension to 2-D feature extension shown in Fig. 2. Four points indexed 1,2,3 and 4 are projected onto a straight line, an ellipse and a hyperbola. As is evident, the interpoint distances in the projected space could be increased substantially by choosing an appropriate function. In addition, such projections alter the continuous probability density function $p(\mathbf{X})$ of the vectorial input pattern $\mathbf{X}$. SOMs trained on these projected input spaces obviously exhibit a modified topology and classification results. This paper aims to demonstrate the advantages of projecting the input feature vectors onto high dimensional spaces for classification task.

2.1 Types of Projections

The classification system designer has numerous options in respect of choosing the hyperspheres and manifolds to enhance the classification task. In this paper, we limit ourselves to three simple types of projections.

Let $X = \{x_1, x_2, \cdots, x_i, \cdots, x_n\}$ be an n-dimensional vector. Let x_{n+j} be the $n+j$ th feature element of the extended feature vector.

Type A: Extension by one dimension:

The feature element x_{n+1} is computed by assuming a projection onto $n+1$ dimensional sphere whose origin is placed at the same location of the origin of n dimensional vector space. Let R be the radius of the sphere. The value of R is chosen to be the maximum Euclidean norm of the input feature vectors. Then

$$x_{n+1} = \sqrt{R^2 - \sum_{i=1}^{n} x_i^2} \tag{1}$$

Type B: Extension to 2n dimensions:

The extended feature vector will have 2n dimensions. Each of its elements in the extended part is

Projection Pursuits in SOM Classifiers *

V. Chandrasekaran†and Zhi-Qiang Liu‡

Department of Computer Science
University of Melbourne,
Carlton, Victoria-3053, Australia
e-mail: †vc@cs.mu.oz.au ‡zliu@cs.mu.oz.au

Abstract

In this paper, we propose a novel approach to enhance the classification performance of Self-Organizing Map based classifiers by extending the input pattern dimensionality. Depending on the technique used in projecting a pattern onto a high dimensional manifold, it is shown that the input patterns that lie very close to each other can be separated. Such projections alters the probability distribution in feature space and hence the topological ordering in SOM as well. The technique is demonstrated on a set of 8-dimensional feature data obtained from 8-class texture images. Kohonen's SOM and its extensions to higher-order correlation maps are tested with the proposed technique and the results indicate a consistent improved performance for a specific type of projection.

1 Introduction

Self-Organizing Maps (SOMs) have been successfully used for a wide range of pattern recognition tasks. The classification performance of such SOMs are usually improved by supervised learning paradigms such as LVQ1 to 3. [3]. There are a number of techniques to enhance the performance of SOM based classifiers. A recent technique by Chandrasekaran et al.[1] is an extension of SOM to a Spatio-Temporal Feature Map (STFM) by introducing a selective competition strategy among the neurons resulting in more input information in the form a winning node sequence to the classifier. This technique, however, suffers from high sensitivity to noise due to extreme levels of feature space fragmentation. Another approach is to use a multiple map integration strategy where predictions from topologically different maps created for the same input space (either using correlated inputs or different metric spaces) are simultaneously considered to predict the input pattern's class [2]. All of the above techniques consider modifications to topological ordering, winner-take-all strategy and/or prediction strategy.

In this paper, we analyze SOM performance by considering input pattern projections onto high-dimensional manifold to provide a better inter-class separability. Such projection pursuit approach has already been successfully implemented in multi-layer feedforward neural networks [4]. The result of projection pursuit implementations in feed forward neural networks have demonstrated reduced network size and many orders of decrease in training times required over conventional BPNN. This has motivated us to investigate the above approach in SOM classifiers.

A general organization of the proposed system is shown in Fig. 1. The classifier output is based on a simple winner-take-all strategy. Each node is assigned a class label based on the maximum class frequency obtained during the training data. When two classes have the same maximum frequency, then one of them is chosen at random as a winning label with others being treated as misclassified.

*This work is supported by Australian Research Council large grants. ‡Address all correspondences to Dr. Zhi-Qiang Liu