NEURAL NETWORK FOR VISUAL SEARCH CLASSIFICATION

H. Raju[1], R.S. Hobson[2], P.A. Wetzel[1]

[1]Department of Biomedical Engineering, Virginia Commonwealth University, VA, USA

[2]Department of Electrical Engineering, Virginia Commonwealth University, VA, USA

Abstract - **Visual search describes the process of how the eyes move in a visual field in order to acquire a target. Visual search needs to be quantified to improve future search strategies. This paper describes a hybrid neural network used to perform visual search classification. The neural network consists of a *Learning vector quantization network (LVQ)* and a *single layer perceptron*. The objective of this neural network is to classify the various human visual search patterns into pre-determined classes. The classes signify the different search strategies used by individuals to scan the same target pattern. The input search patterns are quantified with respect to an ideal search pattern, determined by the user. A supervised learning rule, *Learning vector quantization1* (lvq1) is used to train the network.**

I. INTRODUCTION

The present project relates to a LVQ network for pattern classification, which is able to vary it's response signal by learning to separate and to identify a correct class of the input signal from repeated presentations of an input pattern signal. Vector quantization traditionally is a technique used for compression of speech and image data [7]. The aim of the network is to classify visual search. During free viewing where there is no objective, eye position can be modeled as a Markov chain with stable probabilities to describe the position of the fovea. During timed visual search, eye movements tend to be more structured [6]. People follow a very generalized subject dependent search pattern at first, and then follow a very generalized search pattern not dependent on subject but based on the task at hand [7]. There are varieties of schemes employed, such as horizontal step-down function, processing the display in a column format, spiral in or out etc. In this research, the hybrid neural network consists of a LVQ network that uses a supervised competitive learning scheme and a single layer perceptron that uses the least-mean-square (LMS) learning algorithm. The network is simulated in Matlab using the lvq function.

The goal is to classify, compare and quantify visual search patterns. The work described in this paper is the first step towards reaching that goal.

II. METHODOLOGY

The architecture of a LVQ network is very similar to a self-organizing memory. In both networks each neuron excites itself and inhibits all the other neurons. The neuron (i.e. ith processing element) whose weight vector is closest to the input (i.e. winning neuron) has the largest net input and its output is set to one. All other outputs are set to zero. The network modifies the weight of the winning neuron as a function of the error (i.e. the difference between the input vector $P(n)$ and the weight vector $W_i(n)$) and the learning rate á.

$$W_i\,(n+1) = W_i\,(n) + á\,(P(n) - W_I(n)) \tag{1}$$

W_i is the weight matrix if the i^{th} processing element.
$W_i\,(n+1)$ is new modified weight matrix.
$W_i\,(n)$ is the current unmodified weight matrix.
$P(n)$ is the input vector.
á is the learning rate.

The problems that a self-organizing memory has are the tradeoff between fast learning and stability and the problem of dead neurons. These shortcomings are overcome by using the linear vector quantization (LVQ) learning rule. LVQ is a supervised learning rule. The problem of dead neurons is overcome with the use of a conscience mechanism, and if the winning neuron in the hidden layer incorrectly classifies the current input, the weight vector is moved away from the input vector and the weights of the closest neuron to the input vector are modified to move it towards the input. If input $P(n)$ is classified, correct then,

$$W_i\,(n+1) = W_i\,(n) + á\,(P(n) - W_I\,(n)) \tag{2}$$

whereas, if input $P(n)$ is classified in-correctly then,

$$W_i\,(n+1) = W_i\,(n) - á\,(P(n) - W_I(n)) \tag{3}$$

The LVQ network has an input buffer layer for receiving the input pattern signal vector. The network is provided with intermediate nodes for comparing the input pattern to templates of already learned patterns. These templates may be realized as weights wherein a separate weight is provided for each element of the input pattern. Each template has a class associated with it, and there may be multiple templates sharing a common class. These intermediate nodes produce an output indicative of the similarity of the input pattern to the template.

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
25OCT2001	N/A	-

Title and Subtitle Neural Network for Visual Search Classification	**Contract Number**
	Grant Number
	Program Element Number
Author(s)	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Department of Biomedical Engineering, Virginia Commonwealth University, VA	**Performing Organization Report Number**
Sponsoring/Monitoring Agency Name(s) and Address(es) US Army Research, Development & Standardization Group (UK) PSC 802 Box 15 FPO AE 09499-1500	**Sponsor/Monitor's Acronym(s)**
	Sponsor/Monitor's Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes Papers from the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, October 25-28, 2001, held in Istanbul, Turkey. See also ADM001351 for entire conference on cd-rom.	
Abstract	
Subject Terms	
Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU
Number of Pages 3	

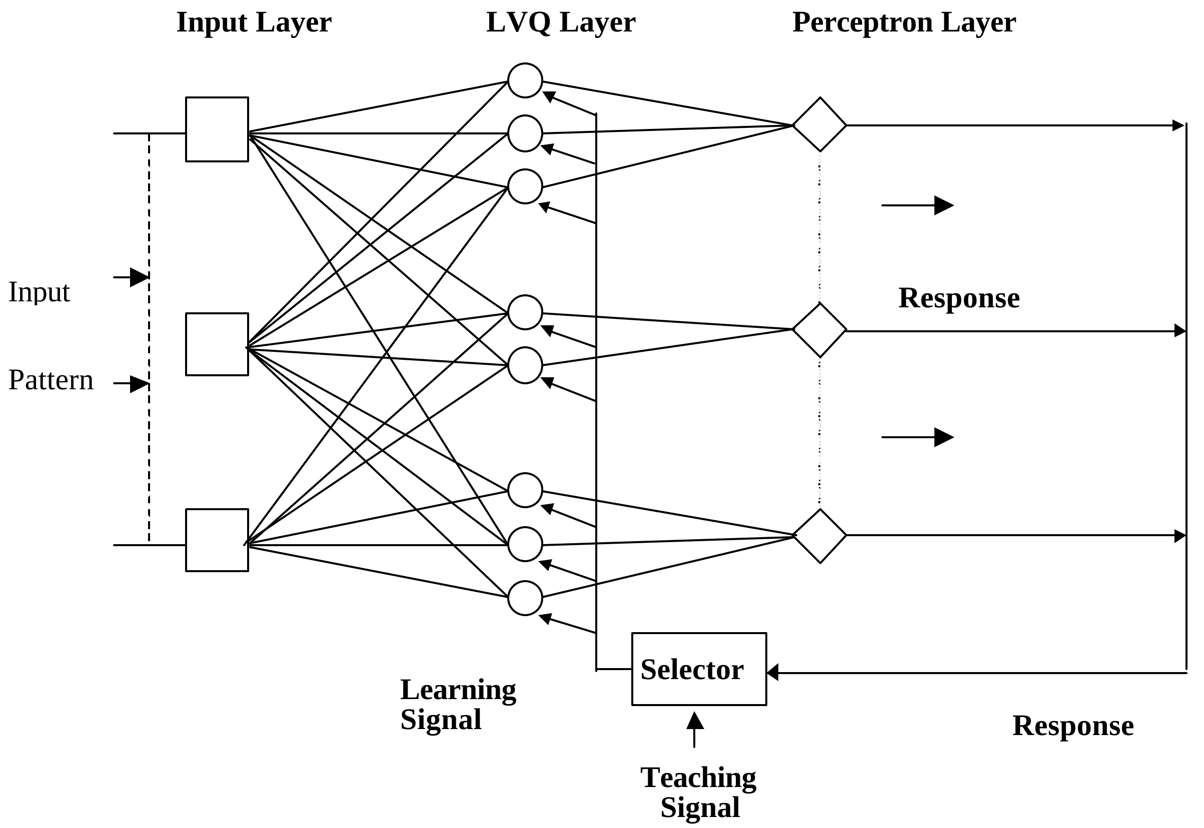

Figure 1. Architecture of the hybrid network.

The similarity between two N-by-1 vectors A and B is defined by,

$$d_{AB} = A - B \tag{4}$$

calculating the Euclidean distance between them. Where d_{AB} corresponds to the euclidean distance. The outputs of the LVQ network are transmitted to the single layer perceptron of the respective class. Each class has a unique processing element in the single layer perceptron associated with it. The single layer perceptron identifies the template that is most similar to the input pattern. The single layer perceptron forwards the node number and the value indicative of the similarity to the input pattern to a selector. The selector chooses the class associated with that template as the class of the input. The selector applies a learning rule lvq1 to the group of intermediate nodes belonging to the selected input class. An overview of the lvq1 algorithm is given below,

1. All the weight vectors and learning rate parameters are initialized.
2. The stopping condition is whether the input pattern is classified or not. If classfication is performed, quit if not, continue.
3. For each input vector the algorithm performs steps 4 and 5.
4. Using equation (4), the weight vector that has a minimum euclidean distance to input vector is selected.
5. The appropriate weight vector is updated using equation (2) or (3) depending on whether the classification is right or wrong.

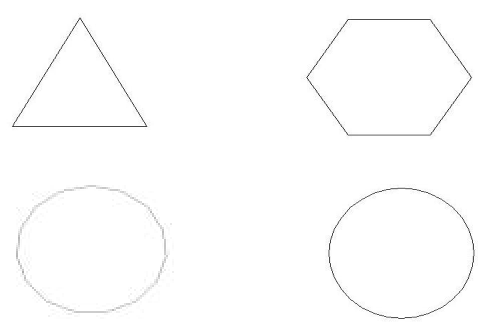

Figure 2. Examples of the polygons with increasing sides

The inputs to the LVQ network were polygons generated using computer aided design software (CAD) as shown in Figure 2. The network was trained on nine polygon inputs composed of 3,4,5,6,7,10,15,20 sides and a circle. These formed the nine unique classes. The polygons were graded depending upon their similarity to the circle. The circle had a radius of 1.5 cm and all the other polygons of the training set are inscribed in the circle. These polygons were converted from the their CAD format to column vectors that could be recognized by the network. The network was trained on these inputs over a period of ten hours and then was able to classify the inputs to the nine unique classes as assigned by the user. This completed the training process of the network.

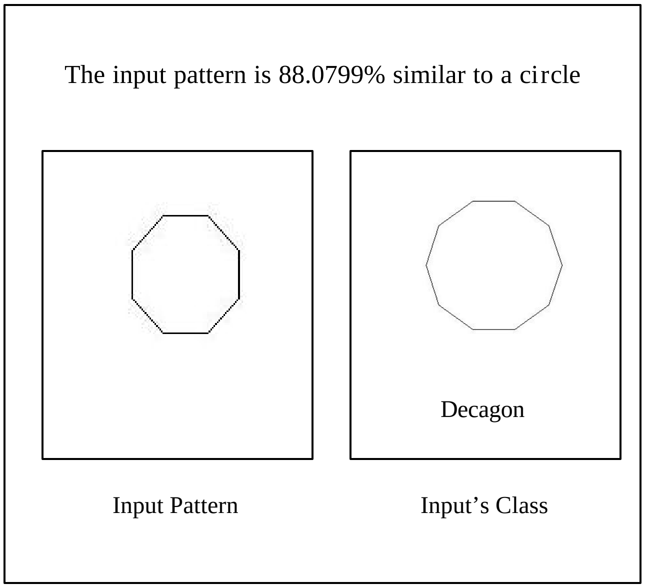

Figure 3. The display format.

The output of the network as shown in Figure 2, was displayed in the form of pop-up figure showing the input pattern, the class it belonged to and the quantified value of its similarity to the circle. The percentage similarity is calculated as a function of the euclidean distance between the two patterns being compared.The display helps the user to see all the results in a single picture.

III. RESULTS

The extended training period of the LVQ network was a result of the size of the input vector. The conversion of the polygon drawings to column vectors resulted in an input vector size of 416,976 by 9. Once the network was trained, it was tested on eleven other polygons, generated in the same manner as the training inputs. When the test input vector was applied to the network, it was placed in a class with the least euclidean distance to the input. The similarity of the input vector to the circle was quantified as a function of the euclidean distance between the two patterns. The similarity gives us a measure to compare the polygons. There were two misclassifications. The polygon of twelve sides was put in class decagon and the polygon of seventeen sides was put in class of polygon of fifteen sides. Hence, the network had an accuracy of eighty two percent. This misclassification was because both the input vector polygons were far apart from any of the classes. This can be overcome by introducing two new classes namely, polygon of twelve sides and polygon of seventeen sides so that the gap between adjacent classes is reduced.

IV. DISCUSSION

The Network described in this paper has not been tested on actual visual search data. The next step will be to modify the network so that it can perform translation and scale invariant pattern classification. The size of the input vector needs to be reduced by only extracting features from the object. Hardware implementation of the LVQ network would improve the performance and operating speed of the network. This work will help in developing a true visual search quantifier and classifier.

V. CONCLUSION

The project provides a neural network self-organizing pattern classification system that is able to properly classify input patterns of data.

REFERENCES

[1] Narayan Srinivasa and Narendra Ahuja, " A topological and temporal correlator network for spatiotemporal pattern learning, recognition, and recall," IEEE Trans. Neural Networks, vol. 10, no. 2, pp. 356-372, March 1999.

[2] Hee-Hong Song and Seong-Whan Lee, "A self-organizing neural tree for large scale pattern classification," IEEE Trans. Neural Networks, vol. 9, no. 3, pp. 369-380, May 1998.

[3] Torbjorn Eltoft and Rui J.P. deFigueiredo, " A self-organizing neural network for cluster detection and labelling," IEEE WCCI, May 4-9th 1998.

[4] Tetsuya Hoya and Jonathan A. Chambers, " Heuristic pattern correction scheme using adaptively trained generalized regression neural networks," IEEE Trans. Neural Networks, vol. 12, no. 1, pp. 91-100, Jan 2001.

[5] Ham, F.M., Kostanic, I., "Principles of Neurocomputing for Science and Engineering," McGraw-Hill, N.Y, NY 2001.

[6] Geri, G.A., Wetzel, P.A., Martin, E.L., "Characteristics of two-dimensional eye and head movements during search of simulated imagery," Perception, 25, 73, 1996.

[7] Enoch, J.M., "Natural tendencies in visual search of complex display," Vision Search Techniques, pp.187-193, 1960.