

CLASSIFICATION OF BREAST CANCER DATABASE USING LEARNING VECTOR QUANTIZATION NEURAL NETWORKS

Alaa Mohamed Elsayad

*Associate Professor, Management Information System Department, King Faisal
University, Saudi Arabia*

Keywords

Learning vector quantization, training algorithm, Wisconsin breast cancer database.

Abstract

Learning vector quantization (LVQ) is a feed forward neural network used for pattern classification. It has a superior performance over back propagation method in the sense of minimizing the classification errors while maintaining rapid convergence. The purpose of this study is to examine the performance of different LVQ learning algorithms on the Wisconsin breast cancer data. Wisconsin database is a well-used database in neural artificial intelligence and machine learning. Four learning algorithms are reviewed and evaluated, LVQ1, LVQ2.1, LVQ3 and OLVQ1. Experimental results are introduced and analyzed including the percentages of true positive, true negative and the prediction rate. LVQ1 has proved slightly better clustering and classification performance than other learning methods.

Introduction

Artificial neural networks (ANNs) have the ability to learn and classify patterns by imitating the human brain learning process. They consist of simple elements called neurons, which are operating in parallel. Connections between neurons have strengths or weights. The connection's weights have been adjusted during the learning process by iteratively comparing the output of the network and the required target. The ability of the network to distinguish between input classes depends on both how good the training algorithm is and to what extent training data is diverse to represent the whole universe of the data.

There are different classes of neural networks, however, most of researches published in the medical studies used one class of neural networks, the back-propagation (BP) [1]. However, several versatile publications have proved that Learning Vector Quantization (LVQ) neural network does better than the back-propagation one in the field of supervised pattern recognition [3,4]. In the cancer research, LVQ has been employed to classify or predict the malignancy for about 15 times in accordance to MEDLINE library. The search was for "LVQ and Cancer" up to December 2006. In 2004 Arun G. Eapen applied LVQ with optimized learning algorithm to predict the malignancy in the Wisconsin breast cancer using LVQ_PAK software [4]. He compared the performance of LVQ with other techniques including decision tree,

association rules and Naïve Bayes. In[5], F. Dieterle et. al., analyzed different markers of breast cancer using LVQ, BP and support vector machine. Other classes of neural network had been applied for breast cancer database in [6]

The main purpose of this study is to analyze the performance of LVQ with different learning algorithms to classify the Wisconsin breast cancer database. The performance of the classification is measured for four different learning algorithms LVQ1, LVQ2.1, LVQ3 and OLVQ1. The database was taken from the University of Wisconsin Hospitals, Madison from Dr. William .H. Wolberg, and available through ftp server [7]. It contains 699 samples with 683 complete data and 16 samples with missing attributes. Each sample contains nine features as tabulated in Table 1. The measurements assigned an integer value between 1 and 10, with 1 being the closest to benign. Class labels benign or malignant are associated for each sample, List of measurement names are shown in Table 1.

Learning vector quantization

Vector quantization (VQ) is a common algorithm in the fields of image and speech processing. Having N data vectors, VQ algorithm groups them into small number of clusters in an unsupervised approach. VQ may be considered as a clustering process. However, LVQ neural network is a supervised classifier first introduced by Kohonen's [2]. It combines clustering and classification processes based on feed forward neural network. Inputs are propagated through a variable number of hidden layers to the output nodes. First, the input data space is partitioned into non-overlapping regions or clusters. Second these regions are mapped to predefined classes. The first step has been accomplished using competitive layer of the network that works similar to the Self-Organizing Map (SOM) [2]. The layer clusters the input data vectors using table of vector prototypes known as codebook. The number of codebook vectors is much less than the number of input data vectors, however, it has to be predefined by the user. In clustering operation, every data vector is assigned to the closest codebook vector according to a predefined distortion measure. The second step of LVQ is accomplished using the linear layer of the network that maps each codebook vector to the target class.

The architecture of Kohonen's Neural network that implements LVQ operations is shown in Fig. 1. It consists of three layers; named *input*, *hidden* called *competitive*, and *output* called *linear* layers. The weights of the input-competitive links represent the codebook vectors. They are M -dimensional vector, as the input vectors, that are positioned in the input data space to identify cluster regions. Clusters borders are defined by a 'Voronoi net' of hyper-planes perpendicular to the linking line of two codebook vectors as shown in Fig. 2. Each neuron in the competitive layer represent ones cluster. The linear layer maps the competitive layer's neurons into target classification defined by the user. Multiple neurons may belong to the same class, however, in the data space, cluster regions corresponding to the same class in the M -dimensional space need not be contiguous. The learning algorithm has to appropriately position competitive 's neurons, codebook vectors, in the M -dimensional

input space and associate them to the right linear neurons, class labels. Different learning algorithms have been proposed, they are all adaptive as the training samples are presented one at a time in random order. The codebook vectors gradually capture the underlying statistical properties of the training data. That is to avoid both the falling in local optima and the complexity of gradient calculation. As a result, LVQ networks are statistical classifiers, which rapidly converge to a good solution.

Design and learning of LVQ

The first step in the design of LQV neural network is setting the parameters of both competitive and linear layers. Then the available input data vectors have to be partitioned into training and test groups. Learning algorithm generally works as follows:

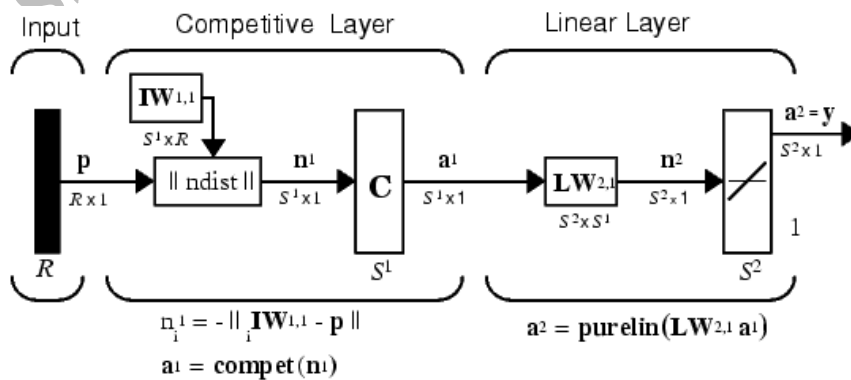
1. Codebook initialization: The number of codebook vectors for each target class has to be comparative to the number of occurrence of that class and these vectors are initialized to the center of the input ranges.
2. Winner determination: The Euclidean distance has to be calculated between training data vector and each codebook vector,

$$d = \|w_j - x_i\| = \sum_i (w_{ji} - x_i)^2 \quad (1)$$

Where x is a training data vector, w is codebook vector. x and w are M -dimensional and d is the Euclidean distance.

The neuron m_c with codebook vector that has the least Euclidean distance to a data vector x_i is considered a winner.

$$m_c = \arg \min_j (d_j) \quad (2)$$



Where...

R = number of elements in input vector

S^1 = number of competitive neurons

S^2 = number of linear neurons

Fig. 1 learning vector quantization neural network architecture [8]

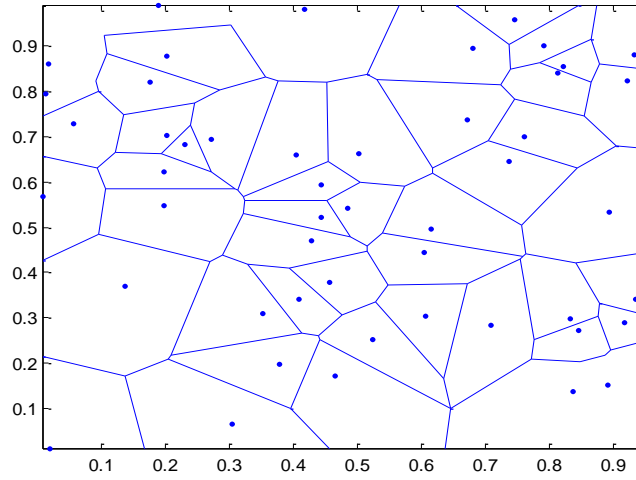


Fig. 2. 2D clusters and cluster centers (Voronoi net)

3. **Codebook Adaptation:** Codebook vectors are optimized during learning process. Different learning algorithms have been proposed. They are all iterative gradient methods. The purpose is to find the optimal codebook and avoid complex gradient calculations. Here the first four methods that published for learning LVQ are reviewed and evaluated as follows:

LVQ1

Kohonan proposed the basic learning vector quantization (LVQ1) in 1989 as a classification algorithm combining vector quantization and supervised learning [2]. LVQ1 starts by randomly selects a training vector x , finds the nearest codebook vector m_c which is called the winner and moves this winning neuron toward the training data vector if both of them belong to the same class, if not the neuron will be moved away and all other neurons are kept unchanged.

$$\begin{aligned} m_c(t+1) &= m_c(t) + s(t)\alpha(t)[x(t) - m_c(t)] \\ m_i(t) &= m_i(t) \quad \text{for } i \neq c \end{aligned} \quad (3)$$

$$s(t) = \begin{cases} +1 & \text{If } L(m_c)=L(x) \\ -1 & \text{otherwise} \end{cases}$$

where $0 < \alpha_c(t) < 1$ is the learning rate, and $L(x)$ is the

label of x .

LVQ 2 and LVQ 2.1

If the training vector x lies near the border of two classes, LVQ2 adjusts the two the codebook vectors of both classes. The original LVQ2 puts more condition that the winning vector m_c has to belong to a different class than x , and the second nearest class m_c^{\setminus} belongs to the same class as x [9]. This algorithm was adapted to LVQ2.1, that requires one of m_c or m_c^{\setminus} should belong to the correct class and the other to an incorrect class.

$$\begin{aligned} m_c(t+1) &= m_c(t) + \alpha(t)[x(t) - m_c(t)] \\ m_c^{\setminus}(t+1) &= m_c^{\setminus}(t) - \alpha(t)[x(t) - m_c^{\setminus}(t)] \\ m_i(t) &= m_i(t) \quad \text{for } i \neq c \end{aligned} \quad (4)$$

Considering d_c to be the distance between the training vector x and m_c , and d_c^{\setminus} is the distance between the training vector x and m_c^{\setminus} , then the training vector x is considered to be in the border if

$$\frac{d_c}{d_c^{\setminus}} > \frac{1-w}{1+w} \quad (5)$$

Where $w \in (0,1)$ is a predefined window width, typical values of w lies between 0.2 and 0.3.

LVQ3

This algorithm was proposed as a combination of LVQ1 and LVQ2.1 [9]. LVQ3 modifies LVQ2 to include the cases where all x , m_c and m_c^{\setminus} belong to the same class. For such cases LVQ3 adaptation role works as follow

$$\begin{aligned} m_c(t+1) &= m_c(t) + \varepsilon\alpha(t)[x(t) - m_c(t)] \\ m_c^{\setminus}(t+1) &= m_c^{\setminus}(t) + \varepsilon\alpha(t)[x(t) - m_c^{\setminus}(t)] \\ m_i(t) &= m_i(t) \quad \text{for } i \neq c \end{aligned} \quad (6)$$

Where $\varepsilon \in (0,1)$ is a constant stabilizing factor

Stabilizing factor is added to ensure that the codebook vectors continue approximating the class vector distributions.

OLVQ1

OLVQ1 is an optimized version of the original LVQ1 by defining individual learning rate $\alpha_c(t)$ for each $m_c(t)$ [10]. The learning rate $\alpha_c(t)$ of the winning vector is updated by the following rule

$$\alpha(t) = \frac{\alpha(t-1)}{1 + s(t) \bullet \alpha(t-1)}$$

(7)

However, it is necessary to set up an upper limit to $\alpha_c(t)$ to not increase above 1.

3. Experimental results

Experiments have been done using learning vector quantization package LVQ_PAK [11]. The package contains the required learning algorithms, LVQ1, LVQ2, LVQ3, and OLVQ1, all of them were applied to the Wisconsin breast cancer database. The database contains 699 cases where 16 cases contain incomplete features, such that only 683 cases out of the database were used. 400 cases were used for training and the rest for test the networks. Each sample contains nine features as tabulated in Table 1.

Table 1. Nine features in Wisconsin breast cancer database

Feature Number	Feature Name	Feature Number	Feature Name
F_1	Lump Thickness	F_6	Bare Nuclei;
F_2	Uniformity of Cell Size	F_7	Bland Chromatin—evaluates for the presence of Barr bodies;
F_3	Uniformity of Cell Shape	F_8	Normal Nucleoli;
F_4	Marginal Adhesion fibrous bands tissue that form between two surfaces	F_9	Mitoses – cell growth.
F_5	Single Epithelial Cell Size – the size of a single cell that forms tissues that lines the outside of the body and the passageways that lead to or from the surface		

The number of competitive neurons is fixed in all networks and equal twelve. They are divided to seven for benign class and five for malignant class according to the distribution of training cases. Learning rate and number of iterations are fixed for all methods also.

Classification results for all methods are shown in Table 2 for training data and Table 3 for test data. Clustering results are shown in Tables 4 and 5 for training and test data respectively. The performance of classification for each LVQ network is evaluated by four values: numbers of true positives, false positives, true negatives and false negatives. Prediction rate, the percentage of true positives and true negatives, is measured and included in Tables 2 and 3 also. The four methods have roughly equal predication rates for both training and test data vectors.

Table 2. Results of the classification of the training data vectors

Method	Training cases				Prediction rate
	288 Benign cases		172 Malignant cases		
	True negatives	False negatives	True positives	False positives	
LVQ1	280	8	166	6	0.965
LVQ2	279	9	171	1	0.975
LVQ3	279	9	170	2	0.9725
OLVQ1	281	7	164	8	0.9625

Table 3. Results of the classification of the test data vectors

Method	Test cases				Prediction rate
	216 Benign cases		67 Malignant cases		
	True negatives	False negatives	True positives	False positives	
LVQ1	213	3	66	1	0.985866
LVQ2	210	6	66	1	0.975265
LVQ3	212	4	66	1	0.982332
OLVQ1	213	3	65	2	0.982332

The efficiency of a clustering algorithm may be measured using different validity measures[13]. However the sum of the distances, between each data vector and the cluster center to which it belongs, could be used as a measure of the clustering efficiency. LVQ competitive clustering is hard one where any data vector belongs to only one cluster and sum of distances is computed as follows:

$$S_d = \sum_{i=1}^c \sum_{k=1}^n u_{ki} (x(k) - v(i))^2, u_{ki} = \begin{cases} 1 & x(k) \in A_i \\ 0 & x(k) \notin A_i \end{cases} \quad (14)$$

Sum of distances results and other clustering statistics are included in Tables 4 and 5. They contain sum of distances, mean, minimum and maximum distance.

Table 4. Clustering results for training data vector

	LVQ1	LVQ2	LVQ3	OLVQ1
Sum of distances	1.40E+03	1.63E+03	1.46E+03	1.39E+03
Mean distance	3.497	4.083	3.6574	3.4755
Min. distance	0.1134	0	0.4749	0.0633
Max. distance	10.6623	11.2771	11.9829	10.6449

Table 5. Clustering results for test data vectors

	LVQ1	LVQ2	LVQ3	OLVQ1
Sum of distances	840.0443	1.05E+03	8.61E+02	849.3417
Mean distance	2.9684	3.7041	3.041	3.0012
Min. distance	0.417	0.9331	0.4749	0.4101
Max. distance	10.3302	11.5129	10.6911	10.4058

Experimental results for test data vectors show that LVQ1 has slightly better classification performance and better clustering with prediction rate 0.986%, percentage of true positive 0.986%, percentage of true negative .985%. LVQ1 and OLVQ1 have almost equal performances, then LVQ3 and finally LVQ2. Although speed of the operation is not included here as a measure for comparison, all four methods are very fast classifying new cases.

REFERENCES

- [1] Dayhoff JE, DeLeo JM., Artificial Neural Networks. Cancer 2001;91:1615-35
- [2] Kohonon T, The self-organizing map, Proceedings of the IEEE, 78(9):1464-1480, 1990.
- [3] Mattfeldt TM, Kestler HA, Hautmann R, Gottfried HW., Prediction of prostatic cancer progression after radical prostatectomy using artificial neural networks a feasibility study. Br J Urol 1999;84:316-23
- [4] Arun G. Eapen, Application of Data mining in Medical Applications, Master thesis,2004, University of Waterloo. etd.uwaterloo.ca/etd/ageapen2004.pdf
- [5] Frank Dieterel, Silvia Muller-Hagedorn Harmut M. Liebich, Gunter Gauglitz, Urinary nucleosides as potential tumor markers evaluated by learning vector quatization, Artificial Intelligence in Medicine, Artificial Intelligence in Medicine, 2003; 28: 267-279.
- [6] Tuba Kiyani, Tulay Yildirim, Breast cancer diagnosis statistical neural networks. Istanbul university, Journal of Electrical & Electronic Engineering, 2004; vol. 4; No. 2:1149-1153.

- [7] Wolberg, W.H., & Mangasarian, O.L. 1990. "Multisurface method of pattern separation for medical diagnosis applied to breast cytology". In Proceedings of the National Academy of Sciences, 87, 9193-9196. [<http://pbil.univlyon1.fr/library/mlbench/html/BreastCancer.html>]
- [8] Howard Demuth, Mark Beale, Martin Hagan, Documentation of Matlab neural network toolbox, software package of Mathworks, www.mathworks.com.
- [9] Kohonen T, Improved versions of learning vector quantization. Processing of the International Joint Conference on Neural Networks, I :545-550, San Diego, June 1990.
- [10] Kohonen T., New developments of learning vector quantization and the self-organizing map, Symposium on Neural Networks; Alliances and Perspectives in Senri 1992 (SYNAPSE' 92), Osaka, JAPAN.
- [11] Teuvo Kohonen, Jussi Hynninen, Jari Kangas, Jorma Laaksonen, and Kari Torkkola LVQ_PAK: The Learning Vector Quantization Program Package. Technical Report A30, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland, 1996.
- [13] A.M. Bensaid, L.O. Hall, J.C. Bezdek, L.P. Clarke, M.L. Silbiger, J.A. Arrington, and R.F. Murtagh. Validity-guided (Re)Clustering with applications to image segmentation. *IEEE Transactions on Fuzzy Systems*, 4:112-123, 1996.