A variant of Learning Vector Quantizer based on split-merge statistical tests

C. Kotropoulos and I. Pitas

Department of Electrical Engineering, University of Thessaloniki, Thessaloniki 540 06, GREECE

1 Introduction

Neural networks (NN) is a rapidly expanding research field which attracted the attention of scientists and engineers in the last decade [1]. A large variety of artificial neural networks has been developed based on a multitude of learning techniques and having different topologies [1,2]. One of the most prominent neural networks in the literature is the Learning Vector Quantizer (LVQ) [3]. It is an associative nearest neighbor classifier which classifies arbitrary patterns into N-many classes using an error-correction learning procedure related to the competitive learning. Among others, it has been applied for image segmentation [4,5] etc. For a complete list of the numerous applications LVQ has found, the interested reader may consult [1,3].

As its name suggests, LVQ is essentially a vector quantization method. A vector quantizer (VQ) maps data from a p-dimensional space to a finite set of points called a codebook, where each point is called a codevector. One common feature of vector quantization methods is that they rely on the assumption that the size of codebook (i.e., the number of ouput neurons) N, is known in advance or is preset to a desired value, usually a power of 2. To the author's knowledge, no attempt has been made to treat thoroughly the problem of optimal N. Although several VQ design techniques employ splitting criteria, the availability of splitting criteria is not sufficient from its own. Since a quantization in terms of $N+1$ output neurons is always expected to yield a lower Mean-Squared Error (MSE) than a quantization in terms of N output neurons [6], the issue of deciding what constitutes a statistically significant improvement in MSE should also be addressed. Another common feature of the bulk of nearest neighbor clustering procedures (like LVQ) is the unconditional inclusion of each training vector provided in the input of VQ in the cluster of the nearest-neighbor codevector (i.e., the nearest-neighbor output neuron), whether the incoming training vector is statistically similar or not to the remaining training vectors that were previously included in the same cluster. Therefore, efficient means for rejecting outliers in the formation of minimum distortion partition of a VQ should be developed.

Motivated by the above described open questions, a split-merge LVQ algorithm is proposed that incorporates statistical hypothesis testing on mean vectors as well as additional tests that are used to determine if cluster splitting/merging is statistically significant. Testing statistical hypotheses on mean vectors [7,8] may serve as an outlier detection mechanism. Cluster splitting is justified by applying tests on the sum of squared-errors. These tests determine if the reduction in the above-mentioned criterion is statistically significant [6,8]. Cluster merging is justified on the grounds of statistical tests that decide if two mean vectors are statistically equal [4,7,8]. The later tests have been discussed thoroughly in [4] and will not described in this paper. Experimental results verify the superiority of the proposed split-merge LVQ algorithm with respect to the number of activated output neurons, the probability of classification of each training vector as well as the MSE after the learning phase has been led to convergence.

2 Split-Merge Learning Vector Quantizer

Learning Vector Quantizer relies on the Euclidean distance in order to determine the best matching output neuron ("winner"). It is well known that in order for the Euclidean distance to be a most effective measure, input patterns must be linearly transformed to another ones that are uncorrelated and of equal variances [8]. Such a linear transformation yields the so-called Mahalanobis distance [7,8], i.e.:

$$\| \mathbf{x}(k) - \mathbf{w}_c(k) \|_M = \left[(\mathbf{x}(k) - \mathbf{w}_c(k))^T \mathbf{S}_c^{-1}(k)\,(\mathbf{x}(k) - \mathbf{w}_c(k))\right]^{1/2} \quad . \tag{1}$$

It is readily seen that by simply replacing the Euclidean distance with the Mahalanobis one in the LVQ encoding algorithm, the inversion of the sample dispersion matrix having dimensions $p \times p$ would be required at each input pattern presentation. Such a modification would result in additional $\mathcal{O}(p^2)$ arithmetic operations (e.g. multiplications/additions) under the best circumstances, i.e., when matrix-inversion lemma [9] could be invoked. It will be seen later on that by employing tests on the mean vectors as an outlier rejection mechanism a quadratic form similar to (1) is needed to be evaluated and to be compared to a threshold. But in the later case, the number of times such an additional computation has to be performed is limited only to: (i) the pattern presentations during the first session (i.e., when the whole training set is presented to the input of LVQ for first time), (ii) the patterns that move from one cluster to another, and, (iii) the patterns of a cluster where a modification (i.e., insertion/removal of a pattern) has occurred during the session that modification took place. By experiments, it has been found that the number of times the above-mentioned test should be applied is much smaller that the total number of iterations needed until convergence.

2.1 Criteria for Detecting Outliers

Let us assume that an arbitrary number of output neurons exists in the output of an LVQ. Let $N(0)$ denote the number of initial neurons. Since all the statistical tests that will be employed next rely on first and second-order statistics, each output neuron evaluates the sample mean vectors and the sample dispersion matrix associated with the cluster it represents. We shall also assume that a counter is associated with each neuron which counts the number of patterns that belong to the cluster represented by that neuron. At the beginning, the sample mean $\mathbf{m}_j$, the sample dispersion matrix $\mathbf{S}_j$ as well as the number of patterns n_j of each neuron's cluster are appropriately initialized.

Let us denote by i the index that counts the training sessions. For each pattern presentation $\mathbf{x}(k)$ $k = 1, \ldots, M$ during the i-th session, the winner neuron is found:

$$\| \mathbf{x}(k) - \mathbf{w}_c^{(i)}(k) \| = \min_{j=1}^{N^{(i)}(k)} \{\| \mathbf{x}(k) - \mathbf{w}_j^{(i)}(k) \|\} \ . \tag{2}$$

$N^{(i)}(k)$ is the number of output neurons when the k-th training vector is presented in the input of LVQ during the i-th training session. Then, the number of patterns that are represented by the winner as well as the sample mean vector of the cluster associated with the winner are updated as if input pattern $\mathbf{x}(k)$ were merged into that cluster, as follows:

$$\begin{aligned} n_c^{(i)}(k) &= n_c^{(i)}(k-1) + 1 \qquad \mathbf{d}_c^{(i)}(k) = \mathbf{x}(k) - \mathbf{m}_c^{(i)}(k-1) \\ \mathbf{m}_c^{(i)}(k) &= \mathbf{m}_c^{(i)}(k-1) + \frac{1}{n_c^{(i)}(k)} \mathbf{d}_c^{(i)}(k) \ . \end{aligned} \tag{3}$$

Such an updating is required during the first training session ($i = 1$) as well as for $i \geq 2$ when $c^{(i)}(k) \neq c^{(i-1)}(k)$, where $c^{(i)}(k)$ denotes the index of the winner neuron at the presentation of the k-th pattern during i-th iteration. This is case (ii) outlined above. Case (iii) refers to the remaining patterns of a cluster which has been modified due to an insertion/removal of another pattern. Since $c^{(i)}(k) = c^{(i-1)}(k)$, for a moment, we exclude pattern $\mathbf{x}(k)$ from the cluster of patterns that is represented by the winner. Our purpose is to test if its inclusion to that cluster is still valid. Therefore, we have:

$$\begin{aligned} n_c^{(i)}(k) &= n_c^{(i)}(k-1) \qquad n_r^{(i)}(k) = n_c^{(i)}(k) - 1 \\ \mathbf{m}_c^{(i)}(k) &= \mathbf{m}_c^{(i)}(k-1) \qquad \mathbf{d}_r^{(i)}(k) = \mathbf{x}(k) - \mathbf{m}_c^{(i)}(k) \\ \mathbf{m}_r^{(i)}(k-1) &= \mathbf{m}_c^{(i)}(k) - \frac{1}{n_r^{(i)}(k)} \mathbf{d}_r^{(i)}(k) \ . \end{aligned} \tag{4}$$

A similar provision has to be made for the sample dispersion matrix of that cluster. Clearly $\mathbf{S}_c^{(i)}(k) = \mathbf{S}_c^{(i)}(k-1)$, and

$$\mathbf{S}_r^{(i)}(k-1) = \frac{n_c^{(i)}(k)}{n_r^{(i)}(k)} \left\{ \mathbf{S}_c^{(i)}(k) - \frac{1}{n_r^{(i)}(k)} \mathbf{d}_r^{(i)}(k) \mathbf{d}_r^{(i)}(k)^T \right\} \ . \tag{5}$$

Obviously, the initail conditions for updating the equations (3)–(5) are:

$$n_c^{(i)}(0) = n_c^{(i-1)}(M) \quad \mathbf{m}_c^{(i)}(0) = \mathbf{m}_c^{(i-1)}(M) \quad \mathbf{S}_c^{(i)}(0) = \mathbf{S}_c^{(i-1)}(M) \tag{6}$$

In the sequel, we define a binary hypothesis testing problem. Under the null hypothesis, H_0, we assume that the mean vector of the cluster represented by the winner neuron, $\underline{\mu}$, equals $\mathbf{m}_c^{(i)}(k)$. Under the alternative hypothesis, H_1, $\underline{\mu}$ equals $\mathbf{m}_\gamma^{(i)}(k-1)$, $\gamma \in \{c, \tau\}$. The covariance matrix Σ is assumed unknown under both hypotheses. Recall that $\mathbf{m}_\gamma^{(i)}(k-1)$ is actually the sample mean of that cluster, $\bar{\mathbf{x}}$, because $\mathbf{m}_c^{(i)}(k)$ has been computed as if $\mathbf{x}(k)$ were merged to the cluster under examination, which is what we are going to test. Let

$$\mathbf{d} = \mathbf{m}_\gamma^{(i)}(k-1) - \mathbf{m}_c(k) = -\frac{1}{n_\gamma^{(i)}(k)} \mathbf{d}_\gamma^{(i)}(k) \quad \gamma \in \{c, \tau\} \ . \tag{7}$$

The null hypothesis is accepted, i.e., we decide that merging $\mathbf{x}(k)$ with the remaining patterns is valid, if [7]:

$$\left(\frac{n_c^{(i)}(k-1) - p}{p} \right) \mathbf{d}^T [\mathbf{S}_c^{(i)}(k-1)]^{-1} \mathbf{d} \leq F_{p, n_c^{(i)}(k-1) - p; 0.05} \quad \text{for cases (i), (ii)}$$

$$\left(\frac{n_\tau^{(i)}(k) - p}{p} \right) \mathbf{d}^T [\mathbf{S}_\tau^{(i)}(k-1)]^{-1} \mathbf{d} \leq F_{p, n_\tau^{(i)}(k) - p; 0.05} \quad \text{for case (iii)} \ . \tag{8}$$

where $F_{p, n-p; 0.05}$ denotes the upper 5% level of significance for the F-distribution with n and $n-p$ degrees of freedom.

If H_0 is accepted, then the winner vector is updated as LVQ suggests [3]:

$$\mathbf{w}_c^{(i)}(k+1) = \mathbf{w}_c^{(i)}(k) + \alpha(i) \left[\mathbf{x}(k) - \mathbf{w}_c^{(i)}(k) \right] \tag{9}$$

where $\alpha(i)$ is a variable adaptation step defined as $\alpha(i) = 0.2 \left[1 - \frac{i}{1000}\right]$ [1,3]. The updating of $n_c^{(i)}(k-1)$ and $\mathbf{m}_c^{(i)}(k-1)$ to $n_c^{(i)}(k)$ and $\mathbf{m}_c^{(i)}(k)$ respectively are assumed valid. Furthermore, the sample dispersion matrix of the cluster associated with the winner is also updated:

$$\mathbf{S}_c^{(i)}(k) = \frac{n_c^{(i)}(k-1)}{n_c^{(i)}(k)} \left\{ \mathbf{S}_c^{(i)}(k-1) + \frac{1}{n_c^{(i)}(k)} \mathbf{d}_c^{(i)}(k) \mathbf{d}_c^{(i)}(k)^T \right\} \tag{10}$$

Moreover, in case (ii) the number of patterns, the sample mean and the sample dispersion matrix associated with the cluster of past winner are corrected. Let $v = c^{(i-1)}(k)$ denote the past winner. Then:

$$n_v^{(i)}(k) = n_v^{(i)}(k-1) - 1 \qquad \mathbf{d}_v^{(i)}(k) = \mathbf{x}(k) - \mathbf{m}_v^{(i)}(k-1)$$

$$\mathbf{m}_v^{(i)}(k) = \mathbf{m}_v^{(i)}(k-1) - \frac{1}{n_v^{(i)}(k)} \mathbf{d}_v^{(i)}(k)$$

$$\mathbf{S}_c^{(i)}(k) = \frac{n_v^{(i)}(k-1)}{n_v^{(i)}(k)} \left\{ \mathbf{S}_v^{(i)}(k-1) - \frac{1}{n_c^{(i)}(k)} \mathbf{d}_v^{(i)}(k) \mathbf{d}_v^{(i)}(k)^T \right\} \ . \tag{11}$$

For the remaining neurons, all the corresponding parameters are left intact, i.e.:

$$n_j^{(i)}(k) = n_j^{(i)}(k-1) \qquad \mathbf{m}_j^{(i)}(k) = \mathbf{m}_j^{(i)}(k-1) \qquad \mathbf{S}_j^{(i)}(k) = \mathbf{S}_j^{(i)}(k-1)$$

$$\mathbf{w}_j^{(i)}(k+1) = \mathbf{w}_j^{(i)}(k) \tag{12}$$

where $j = 1, \ldots, N^{(i)}(k)$, $j \neq c$ in cases (i), (iii) and $j \neq c, v$ in case (ii). In the following, we shall consider what happens when H_0 is rejected.

2.2 Splitting Criteria

If H_0 is rejected, it is reasonable to examine whether the cluster represented by the winner neuron can be split into two subclusters. Let us denote that cluster by $C_c^{(i)}(k-1)$. We shall borrow from the field of cluster analysis [6,8] a statistic that relies on the sum of squared errors $J_e(g)$, $g = 1,2$ to test the validity of the following possibilities: (a) cluster $C_c^{(i)}(k-1)$ is kept united ($g = 1$), and, (b) cluster $C_c^{(i)}(k-1)$ is subdivided into two clusters ($g = 2$), say $C_\zeta^{(i)}(k-1)$ and $C_\eta^{(i)}(k-1)$.

Let us define first the sum of squared errors in cases (a) and (b) outlined above. We have:

$$J_e(g) = \begin{cases} \sum_{\mathbf{x}_j \in C_c^{(i)}(k-1)} \| \mathbf{x}_j - \mathbf{m}_c^{(i)}(k-1) \|^2 & \text{for } g = 1 \\ \sum_{\gamma \in \{\zeta,\eta\}} \sum_{\mathbf{x}_j \in C_\gamma^{(i)}(k-1)} \| \mathbf{x}_j - \mathbf{m}_\gamma^{(i)}(k-1) \|^2 & \text{for } g = 2 \end{cases} \tag{13}$$

where $\mathbf{m}_\zeta^{(i)}(k-1)$ and $\mathbf{m}_\eta^{(i)}(k-1)$ denote the sample mean vectors of the resulted subclusters. In the sequel, we shall describe how the tentative splitting is performed.

We determine the direction in which cluster $C_c^{(i)}(k-1)$ variation is greatest. This amounts to finding the principal component of the sample dispersion matrix (i.e., the eigenvector that corresponds to the largest eigenvalue of $\mathbf{S}_c^{(i)}(k-1)$). Let us denote by $\mathbf{e}_c^{(i)}(k-1)$ the principal (normalized) eigenvector of $\mathbf{S}_c^{(i)}(k-1)$. Having determined $\mathbf{e}_c^{(i)}(k-1)$, we examine the splitting of cluster $C_c^{(i)}(k-1)$ with a hyperplane which is perpendicular to the direction of $\mathbf{e}_c^{(i)}(k-1)$ and passes through the sample mean $\mathbf{m}_c^{(i)}(k-1)$. Therefore, all patterns in $C_c^{(i)}(k-1)$ are sorted into sets $C_\zeta^{(i)}(k-1)$ and $C_\eta^{(i)}(k-1)$ as follows:

$$\begin{aligned} C_\zeta^{(i)}(k-1) &= \left\{\mathbf{x} \in C_c^{(i)}(k-1): \ \mathbf{e}_c^{(i)}(k-1)^T\mathbf{x} \leq \mathbf{e}_c^{(i)}(k-1)^T\mathbf{m}_c^{(i)}(k-1)\right\} \\ C_\eta^{(i)}(k-1) &= \left\{\mathbf{x} \in C_c^{(i)}(k-1): \ \mathbf{e}_c^{(i)}(k-1)^T\mathbf{x} > \mathbf{e}_c^{(i)}(k-1)^T\mathbf{m}_c^{(i)}(k-1)\right\} \ . \end{aligned} \tag{14}$$

As mentioned earlier, splitting of any cluster to two subclusters will result a lower sum of squared errors, i.e., $J_e(2) < J_e(1)$. We decide to consider as valid any splitting that yields a statistically significant improvement (i.e., decrease) in the above-mentioned criterion. To this end, a binary hypothesis testing problem is formulated as follows [6]. Under the null hypothesis we assume that there is exactly one cluster present. Furthermore, it is assumed that all $n_c^{(i)}(k-1)$ patterns come from a multivariate distribution with mean $\underline{\mu}$ and covariance matrix $\sigma^2\mathbf{I}$. Then, $J_e(1)$ is argued that is approximately normal with mean $n_c^{(i)}(k-1)p\sigma^2$ and variance $2n_c^{(i)}(k-1)p\sigma^4$ [6]. Next, the sampling distribution for $J_e(2)$ is computed under the null hypothesis. This distribution expresses what kind of apparent improvement to be expected when the one cluster partition is actually correct. For the splitting provided by a hyperplane through the sample mean and large $n_c^{(i)}(k-1)$, $J_e(2)$ is again approximately normal with mean $n_c^{(i)}(k-1)(p-\frac{2}{\pi})\sigma^2$ and variance $2n_c^{(i)}(k-1)(p-\frac{8}{\pi^2})\sigma^4$ [6].

The null hypothesis is rejected, therefore, splitting is accepted at the ρ-percentage significance level, if

$$\frac{J_e(2)}{J_e(1)} < 1 - \frac{2}{\pi p} - \beta\sqrt{\frac{2(1-\frac{8}{\pi^2 p})}{n_c^{(i)}(k-1)p}} \tag{15}$$

where $\rho = 100\int_\beta^\infty \frac{1}{\sqrt{2\pi}}\exp[-\frac{u^2}{2}]\,du$.

If cluster splitting is accepted, we proceed to the evaluation of the sample dispersion matrices for $C_\zeta^{(i)}(k-1)$ and $C_\eta^{(i)}(k-1)$. Next, it is examined if the current training pattern $\mathbf{x}(k)$ can be merged with one of the subclusters $C_\zeta^{(i)}(k-1)$ or $C_\eta^{(i)}(k-1)$. The following two cases are considered:

1. If $\mathbf{e}_c^{(i)}(k-1)^T\mathbf{x}(k) \leq \mathbf{e}_c^{(i)}(k-1)^T\mathbf{m}_c^{(i)}(k-1)$, possible inclusion of $\mathbf{x}(k)$ in $C_\zeta^{(i)}(k-1)$ is tested by applying the statistic described in Sect. 2.1.
2. Otherwise, possible inclusion of $\mathbf{x}(k)$ in $C_\eta^{(i)}(k-1)$ is tested by applying the statistic described in Sect. 2.1.

In either case, if the null hypothesis of Sect. 2.1 is accepted, the number of patterns, the sample mean and the sample dispersion matrix of the subcluster where $\mathbf{x}(k)$ is merged are appropriately updated. The corresponding parameters of the other cluster are left intact. Moreover, the winner neuron is replaced by the two newly created ones. Their weight vectors are set equal to the sample mean vectors $\mathbf{m}_\zeta^{(i)}(k)$ and $\mathbf{m}_\eta^{(i)}(k)$.

If $\mathbf{x}(k)$ cannot be merged with any of the subclusters created by splitting $C_c^{(i)}(k-1)$, a third subcluster is formed having seed $\mathbf{x}(k)$. In that case, the winner neuron is replaced by three new neurons, i.e., the two products of cluster splitting and a third neuron whose weight vector is set to $\mathbf{x}(k)$.

Finally, we describe what happens when cluster splitting is not accepted, i.e., when (15) does not hold. In the later case, the winner neuron is kept united and an additional neuron is formed corresponding to a distinct cluster having seed $\mathbf{x}(k)$.

The procedure described so far is applied for each training pattern presentation. When the training set has been exhausted, the integrity of the cluster associated with each output neuron is tested once more by applying the splitting criterion described above. Having completed the later test, we compute the average distortion (e.g. MSE) at the end of session i as follows:

$$D(i) = \frac{1}{M}\sum_{k=1}^{M} \| \mathbf{x}(k) - \mathbf{w}_{c(k)}^{(i)}(M) \|^2 \quad . \tag{16}$$

If $(D(i-1) - D(i))/D(i) > \epsilon = 0.001$, we proceed to an additional training session.

3 Experimental Results

The description of one experiment that demonstrates the superiority of the proposed split-merge LVQ algorithm is only included. Four distinct bivariate normal populations have been created. Each one has 1000 2-d patterns (i.e., points). The statistical description of the created populations follows:

$\mathcal{P}_1$ is a set of 2-d patterns distributed according to $\mathcal{G}(10.0, 20.0; 0.64, 1.0; 0.8)$
$\mathcal{P}_2$ is a set of 2-d patterns distributed according to $\mathcal{G}(20.0, 20.0; 1.33, 1.0; 0.5)$
$\mathcal{P}_3$ is a set of 2-d patterns distributed according to $\mathcal{G}(23.0, 16.0; 1.8, 1.2; 0.0)$
$\mathcal{P}_4$ is a set of 2-d patterns distributed according to $\mathcal{G}(10.0, 10.0; 1.0, 1.0; 0.7)$

where $\mathcal{G}(\mu_1, \mu_2;\ \sigma_1, \sigma_2;\ r)$ denotes a bivariate normal distribution. Parameters μ_i and σ_i, $i = 1, 2$ are the expected values and the standard deviations along each dimension respectively, and r denotes the correlation coefficient. The "contours of equal concentration" for probabilities 0.9 and 0.99 of the populations $\mathcal{P}_2$ and $\mathcal{P}_3$ can be found in Fig. 1.

The performance of a modified LVQ algorithm that implements the proposed split-merge criteria has been tested against that of a standard LVQ. The modified LVQ network has two output neurons initially. The number of output neurons for standard LVQ neural network is set to five. It has been found by experiments that increasing the number of output neurons for standard LVQ NN more than 5 does not alter the results obtained. Both neural networks have been trained by the same set. The training set is formed by selecting randomly 10% of the patterns that belong to each population. Three training sessions are required for both NN to converge. At the end of the learning phase, the modified LVQ NN results in five output neurons. Each neuron is associated with a cluster where all training patterns that come from a distinct population have been included. In other words, although the training is unsupervised, we have obtained perfect classification. On the contrary standard LVQ results in three activated output neurons. In Table 1, the learning and the recall MSE are summarized. Information related to the learning phase of split-merge LVQ can be found overlaid in Fig. 1.

Table 1. Learning and Recall Mean-Squared Error

Mean-Squared Error	Split-merge LVQ	Standard LVQ
Learning phase	2.530366	5.599298
Recall phase	2.659161	5.749121

References

1. P.K. Simpson, *Artificial Neural Systems*. Pergamon Press, 1990.
2. Special issue on neural networks, vol. I: Theory & modeling, vol. II: Analysis, techniques & applications, *Proceedings of the IEEE*, vol. 78, no. 9-10, pp. 1409–1680, September-October 1990.
3. T.K. Kohonen, "The Self Organizing Map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, September 1990.

4. C. Kotropoulos, E. Augé and I. Pitas, "Two-Layer Learning Vector Quantizer for Color Image Quantization," in *Signal Processing VI: Theories and Applications* (J. Vandewalle, R. Boite, M. Moonen and A. Oosterlinck, eds.), pp. 1177–1180, Elsevier, 1992.
5. C. Kotropoulos, X. Magnisalis, I. Pitas and M.G. Strintzis, "Nonlinear Ultrasonic Image Processing Based on Signal-Adaptive Filters and Self-Organizing Neural Networks," *IEEE Trans. on Image Processing*, accepted for publication.
6. R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*. New York: J. Wiley, 1973.
7. K.V. Mardia, J.T. Kent and J.M. Bibby, *Multivariate Analysis*. London: Academic Press, 1979.
8. G.A.F. Seber, *Multivariate Observations*. New York: J. Wiley, 1984.
9. A. S. Householder, *The Theory of Matrices in Numerical Analysis*. Mass.: Blaisdell Waltham, 1964.

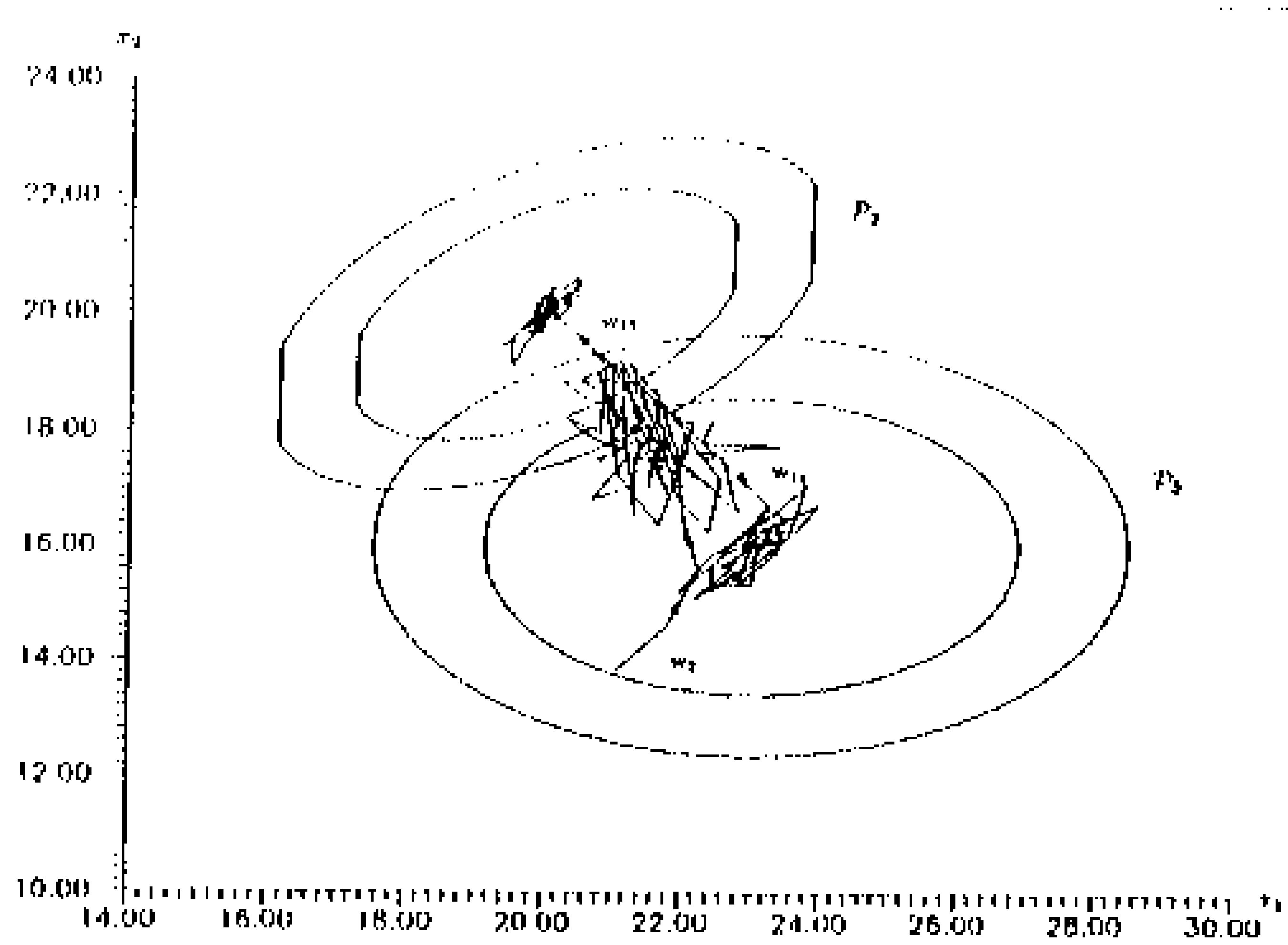

Fig. 1. Contours of equal concentration for probabilities 0.9 and 0.99 of the populations $\mathcal{P}_2$ and $\mathcal{P}_3$. The trajectories of output neuron weight vectors are overlaid.