# STABILITY MEASUREMENT CRITERION FOR NEURAL NETWORKS OF COMPETITIVE LEARNING

AmparoSacristán(*), Elena Valderrama(**), Conrado Pérez-Vicente(**)

(*) Dpto. de Electrónica de la Escuela Universitaria Politécnica de Mataró adscrita a la Universidad Politécnica de Cataluña.

(**) Centro Nacional de Microelectrónica/Dpto. de Informática de la Facultad de Ciencias de la Universidad Autónoma de Barcelona.

**Abstract:** In this article a convergence criterion for neural networks of competitive learning alternative to that established by Rumelhart et al. [Rumel86] is presented. With it the number of iterations needed so that the network reaches a stable configuration is reduced to a high degree. The new convergence criterion is based on the network stability measurement in contrast to the weight variation which is defined by Rumelhart et al. Results obtained in a number of realized tests which allow you to evaluate the step number reduction reached when applying the new criterion as contrasted with Rumelharts are shown.

## INTRODUCTION

Artificial neural networks are able to modify their own behaviour on reply to the inputs shown to them. If you show them a set of inputs and, eventually, the outputs you wish, the network is able to change its structure to make consistent answers. The particular way on which the framework modifies its structure according to the received data is known as *learning rule*.

Neural network operation is determined by its own structure and weights associated to connections, but it is not ruled by any traditional "programm" or "algorithm", as we are used to seeing on Computering and other similar subjects.

One of the points on which the study of neural networks is based is the wish to shape the human brain working. From this point of view, unsupervised learning neural networks (without a teacher guiding the learning) are fairly interesting becouse they can organize themselves before stimuli. This kind of nets becomes stable when the learning algorithm does not change any value of the synapses weights. Due to fact that there is not a "teacher", they are usually networks requiring a high number of iterations to become stable states.

Its upon competitive learning where we have centered studies. After repeating some of Rumelhat's paradigms we tried to find a convergence criterion which will allow to noticeably reduce the excesive number of iterations which his learnig process requires, without changing the configuration of the final network. This new criterion, cause of this work, is based on network stability measurement, parameter introduced by Rumelhart.

This work is divided in three parts. On the first one we shortly present the competitive learning network, as it has been defined by its authors. On the second one, an alternative convergence criterion is shown. And on the third one we study the reduction of the number of iterations which is achieved by our criterion on several practical cases.

## PART I: COMPETITIVE LEARNING NETWORK

**1.1.- DEFINITION:** Competitive learning network is a multilayer neural network. Features:

**1.-ARCHITECTURE.** It is a multilayer neural network with hierarchical structure (see figure 1). Each neuron of on i layer is connected to all the neurons of the i+1 layer. Neurons on the same layer are grouped in "clusters". Only one neuron on each cluster can be active. All the neurons on a cluster receive the same inputs.

**2.- STATES.** Neurons can take the states 0 or 1. Aneuron is said to be active when it is in state 1, and inactive in state 0.

**3.- OUTPUTS.** The output of a neuron coincides with its state.

**4.- SYNAPSES.** Neuron are connected through synapses. A wheight which can take any value between 0 and 1 is associated to each synapses. For all neurons i, the sum of the wheights of their synapses is 1.

$$\Sigma_j \, w_{ij} = 1$$

where $w_{ij}$ is the weight of the synapses which connects neuron j with neuron i.

**5.- PROPAGATION RULE.** The contribution of the network neurons to a neuron i is given by

$$net_i = \Sigma_j w_{ij} \, o_j$$

where $o_j$ is the neuron j output.

**6.- ACTIVATION RULE.** A neuron becomes active if it wins. A neuron i wins when $net_i > net_j$ for all the neurons, on the same cluster. All non-win neuros remain inactive.

**7.- INPUTS.** The neural network accepts $S_k$ vectors (patterns) in the way $(s^1, ..., s^n)$ with $s^i$ {0,1} as inputs.

**1.8.- LEARNING RULE.** For any $S_k$ input stimulus, winning neurons learn by dispaling part of the weights of the synapses coming from inactive neurons to the synapses coming from active neurons. Hence,

$$\text{(1)} \qquad \Delta(w_{rj}(k)) = \begin{cases} g_w(c_{rk}/n_k) - g_w w_{rj}(k), & \text{if } u_j \text{ wins with } S_k \\ g_l(c_{rk}/n_k) - g_l w_{rj}(k), & \text{if } u_j \text{ loses with } S_k \end{cases}$$

where $\Delta(w_{rj}(k))$ represents the increase of r-j synapses weight, with $c_{ik} = 1$ if the i-th component of $S_k$ is 1, and $c_{ik} = 0$ on the other case; $n_k$ is the number of $S_k$ components equal to 1; $g_l$, $g_w$ are the rate of displaced weight for the neuron which lose and win, respectively. It is clearly, them, an unsupervised learning rule.