4

WINNER-TAKE-ALL NETWORKS AND MULTI-CATEGORY CLASSIFIERS

Vasant Honavar

Artificial Intelligence Research Group
Department of Computer Science
Iowa State University
Ames, Iowa 50011

1 INTRODUCTION

So far, we have considered the use of threshold neurons as 2-category pattern classifiers. Many practical pattern classification tasks involve multiple categories. A simple-minded extension of 2-category classifiers to M category classifiers involves the training of each of the M neurons to separate the patterns corresponding its assigned category from the rest of the patterns in the training set using linear hyperplanes. However, as we shall see shortly, this falls short of exploiting the full computational capabilities of a group of M threshold neurons.

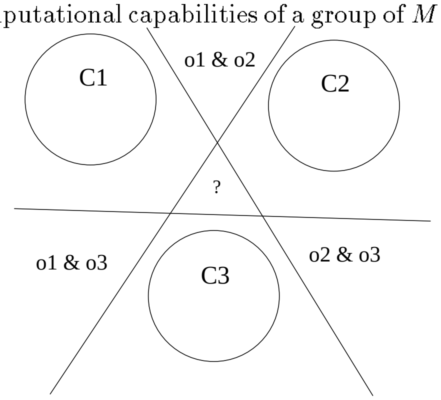

Figure 1 An M-category classifier that uses M independent 2-category perceptrons

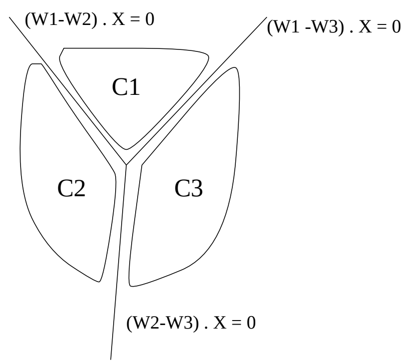

Figure 2 An M-category classifier that uses an M-neuron WTA group

2 WTA NETWORKS

In a WTA network, each of the neurons computes its net input as the dot product of its weight vector with the input pattern. The neuron with the highest net input outputs a 1, and all other neurons output 0. If there is a tie, all of the neurons output 0. This mode of operation is inspired by cortical circuits of neurons which laterally inhibit other neurons in their neighborhood. This inhibition is thought to play a role in a variety of functions from contrast enhancement of visual input to learning. We will examine the detailed processes underlying such competitive interactions among neurons and their implications in terms of information processing functions of the nervous systems later. For now, a simplified algorithmic description of the result of such interaction as outlined above is adequate for our purposes.

Consider a simple network of 3 neurons shown in the figure.

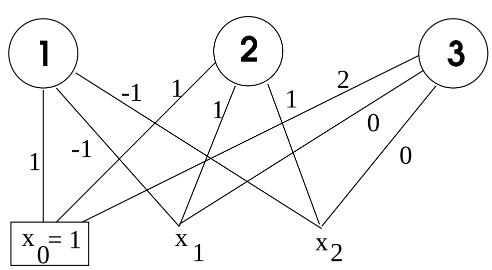

Figure 3 An Example of WTA Network

$\mathbf{W_1}$ =[1 -1 -1]
$\mathbf{W_2}$ =[1 1 1]
$\mathbf{W_3}$ =[2 0 0]

x_1	x_2	x_3	$\mathbf{W_1 \cdot X}$	$\mathbf{W_2 \cdot X}$	$\mathbf{W_3 \cdot X}$	O_1	O_2	O_3
1	-1	-1	3	-1	2	1	0	0
1	-1	+1	1	1	2	0	0	1
1	+1	-1	1	1	2	0	0	1
1	+1	+1	-1	3	2	0	1	0

Upon closer scrutiny, we see that the third neuron (with output O_3) is computing the exclusive-OR function. Of course, we know that exclusive-OR function is not a threshold function. Thus, this example establishes that WTA networks can compute a richer class of functions than a single threshold neuron. We will see that they still fall short of the computing abilities of arbitrary networks of threshold neurons, but nevertheless, they offer interesting designs for multi-category pattern classifiers. We can define the class of functions computed by the WTA groups (linearly separable functions) as follows:

Let $S = \{\mathbf{X_1} \ldots \mathbf{X_p}\}$ be a set of pattern vectors.

Suppose each $\mathbf{X_k} \in S$ belongs to exactly one of M classes: $C_1 \ldots C_m$. Where $S = C_1 \cup C_2 \cup \ldots \cup C_m$, $C_i \cap C_j = \emptyset \ \forall i \neq j$.

The set S is said to be *linearly separable* iff $\exists$ weight vectors $\mathbf{W_1} \ldots \mathbf{W_m}$ such that $\forall$ classes C_j, we have $\forall \mathbf{X}_k \in C_j$, $\mathbf{W_j} \cdot \mathbf{X_k} > \mathbf{W_i} \cdot \mathbf{X_k} \ \forall i \neq j$.

Note that 2-neuron WTA groups are computationally equivalent to a single threshold neuron. Thus, WTA groups are interesting only when the number of neurons in the group is at least 3.

3 TRAING WTA NETWORKS

The learning algorithm for WTA networks is similar to Perceptron learning. Let: $\mathbf{W_i}$ = weight vector of neuron i; $\mathbf{X_k}$ = k^{th} input pattern, $X_{k0} = 1$; O_{ik} = output of neuron i for pattern $\mathbf{X_k}$; $O_{ik} = 1$ iff $\mathbf{W_i} \cdot \mathbf{X_k} > \mathbf{W_j} \cdot \mathbf{X_k} \ \forall j \neq i$, otherwise $O_{ik} = 0$.

As in the case of single neuron training, we cycle through the patterns in the training set one at a time until all the patterns are correctly classified by the winner take all group. However, we modify the weights of the wrongly off neuron and the wrongly on neuron and leave the rest of the weight vectors unchanged. (This is necessary to guarantee that a solution will be found if one exists).

Suppose for the input pattern $\mathbf{X}_k$ belonging to class C_j, the output $O_{ik} = 1$. We want O_{jk} to be a 1, so neuron i is wrongly on and neuron j is wrongly off. We update the corresponding weight vectors as follows:

$$\mathbf{W_i} \leftarrow \mathbf{W_i} - \eta \mathbf{X_k}$$
$$\mathbf{W_j} \leftarrow \mathbf{W_j} + \eta \mathbf{X_k}$$

Note that all other weight vectors $\mathbf{W_l}(l \neq i; l \neq j)$ are left unchanged.

In the event of a tie, we add a fraction of the pattern vector to the weight vector of the neuron that was wrongly off.

Note that there are clearly other ways to solve an M-category pattern classification task. For instance, we can transform the problem into one of solving M 2-category classification problems by attempting to separate each class from the rest using single neuron training algorithm. However, this solution has the following drawbacks relative to WTA training:

- There may be regions in the pattern space for which classification is not unique (where more than one neuron outputs 1).
- There are situations where such a strategy fails to find appropriate weight vectors for a WTA group even if they exist.

Thus it is advantagous to use WTA groups for Multi-category pattern classification.

4 WTA CONVERGENCE THEOREM

If a given multi-category pattern set $\mathbf{S}$ is linearly separable then the WTA algorithm will find the weight vectors $\mathbf{W}_1^* \ldots \mathbf{W}_M^*$ that correctly classify $\mathbf{S}$ using a WTA network with the correspondly weight vectors.

<u>Proof Sketch</u>

Suppose $\mathbf{X} \in \mathbf{S}$ and $\mathbf{X} \in \mathbf{C}_i$. So we want:

$$(\mathbf{W}_i - \mathbf{W}_j) \cdot \mathbf{X} > 0 \forall j \neq i \qquad \boxed{1}$$

$\boxed{1}$ can be written as:

$$\mathbf{W} \cdot \mathbf{P}_{ij} > 0 \forall j \neq i \qquad \boxed{2}$$

where $\mathbf{W} = [\mathbf{W}_1 \ldots \mathbf{W_j} \ldots \mathbf{W_m}]$ is obtained by concatenating the individual weight vectors and $\{\mathbf{P}_{ij}\}$ $(j \neq i,\ 1 \leq i, j \leq M)$ denotes a modified training set constructed as explained below.

Consider a pattern $\mathbf{X}$ that belongs to C_1. Then we construct $(M-1)$ modified training patterns as follows:

$$\mathbf{P}_{12} = [\mathbf{X}\ \ -\mathbf{X}\ \ \phi\ \ \phi \ldots \phi].$$
$$\mathbf{P}_{13} = [\mathbf{X}\ \ \phi\ \ -\mathbf{X}\ \ \phi \ldots \phi].$$
$$\mathbf{P}_{1M} = [\mathbf{X}\ \ \phi \ldots \phi\ \ -\mathbf{X}].$$

where ϕ denotes a vector of zeros (with the same dimension as $\mathbf{X}$. Thus, there are $M-1$ patterns in the modified training set for each pattern in the original training set. It is easy to see that the the original M-neuron WTA group training problem has a solution if and only if the single neuron training problem has a solution. That is, $\boxed{2}$ is a single neuron (perception) training problem with weight vector $\mathbf{W}$ and a modified training set made of $\{P_{ij}\}$. $\boxed{2}$ has a solution iff $\boxed{1}$ has a solution. So the convergence of the WTA training algorithm follows from the perceptron convergence proof.

Note that the transformation of the training set outlined above is not a practical approach to finding the weight vectors for the WTA group (since it requires processing $M-1$ modified patterns for each pattern in the given training set).