

Committee Networks

What They Can and Cannot Do

R. J. Brown Elijah Laboratories Inc
201 West High Street P. O. Box 833
Warsaw KY 41096 (606) 567-4613

The committee network is a 2-layer organization of neurons that has special appeal in certain areas of pattern recognition. It has a binary output, is easy to implement, trains quickly, and has fast response due to only 2 layers of cells. The usual activation function for the neurons in a committee network is a simple switching function, making electronic hardware implementations inexpensive and straightforward. The input pattern may be composed of either discrete binary or continuous analog signals.

When a neuron has a linear input function and a switching activation function, it is called a threshold logic unit. The input function may be viewed mathematically as the dot product of the augmented input pattern vector with the weight point of the neuron. This weight point is just the vector formed from the scalar weighting coefficients at each of the inputs. The input pattern is augmented with a constant non-zero reference value (usually=1) so that the condition can never arise where all the inputs to the neuron would be zero simultaneously. The weight associated with this last input sets the firing threshold of the neuron.

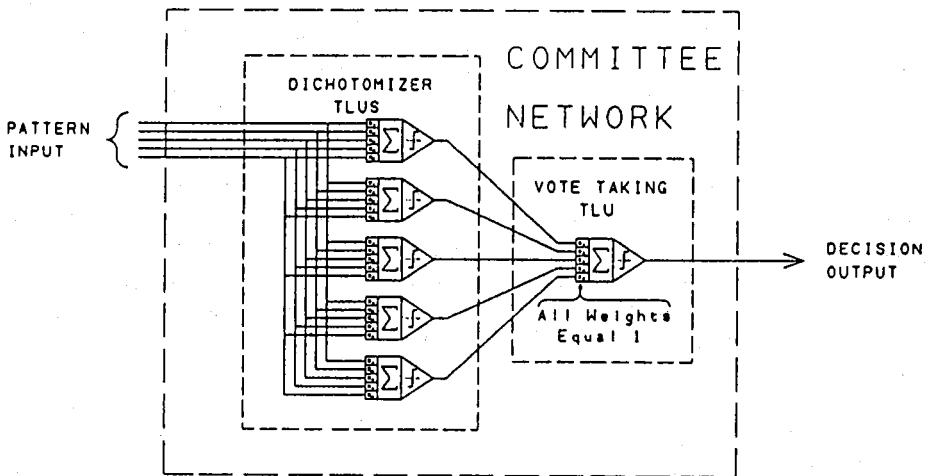
Each layer-1 TLU outputs a +1 or a -1 which is then fed into a single layer-2 TLU that acts as the committee chairman and counts the votes, but does not actually vote itself. The weight point of this chairman TLU is a simplex: all of its weights are set to unity, and are not altered during training. The output of the chairman TLU is the decision of the committee. A set of n such committees can partition a pattern set into 2^n categories.

When suitably trained, such a network serves as a simple pattern recognizer. Since the training procedure achieves its effect by adjusting the weights of the layer-1 TLUs, the training procedure is a linear iterative operation. The training pattern set is presented to the network repeatedly along with the desired output. The training algorithm adjusts the weight points by moving them perpendicular to the pattern plane presented until a majority of weight points lie on the proper side of each pattern plane in the training set. When this has been accomplished, a simple majority vote will always produce the required recognition.

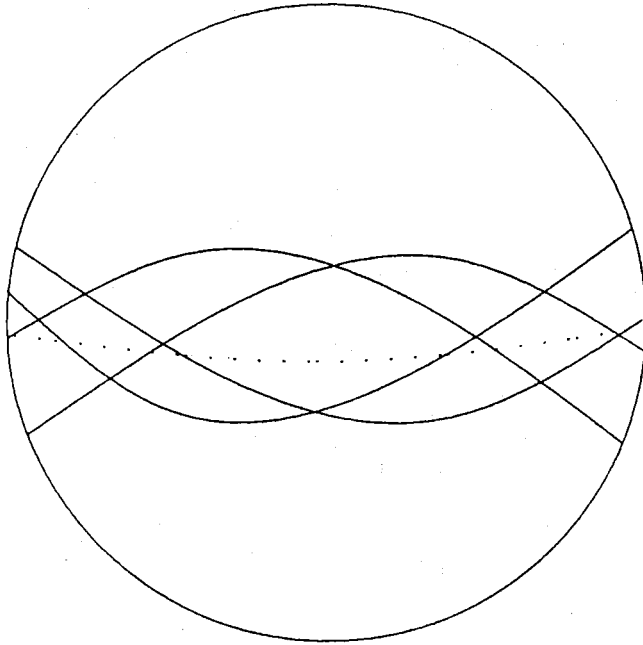
But sometimes the training procedure fails to converge. When the input pattern set is a binary pattern (one whose elements are either 1 or 0), the presence of the zeros causes the corresponding weights to have no effect. This can actually be beneficial if the pattern sets contain a lot of "don't care" situations. This is usually the case for optical character recognition, or manufacturing robot vision systems with non-overlapping components. In other applications, troublesome situations arise that cause convergence failure. If the trouble occurs in a "poly-unsaturated" training set (one

with more don't care cases than interesting cases) using binary input, a few judiciously placed inverters to permute the input will solve the problem, but if the training set is nearly saturated, the network may be unable to learn it.

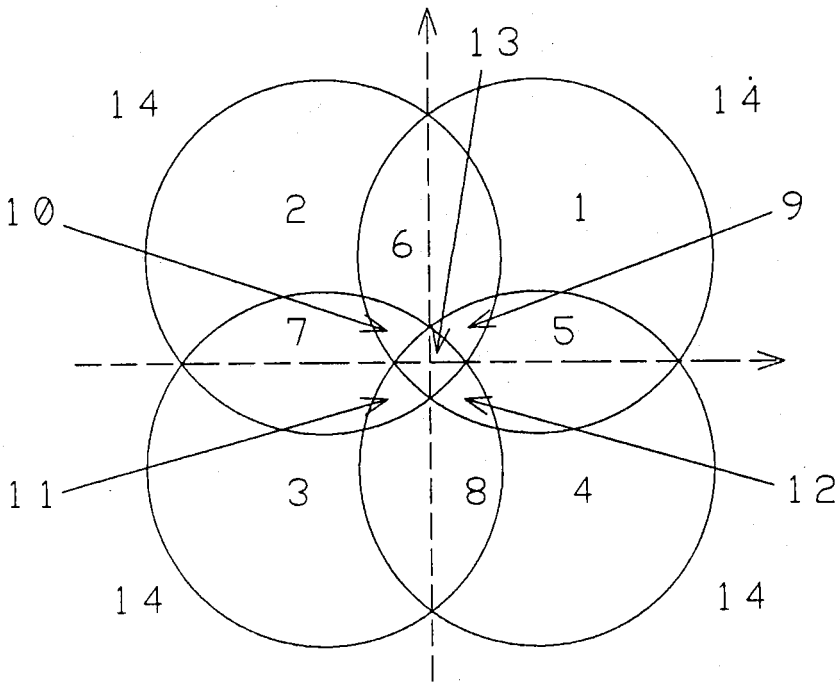
Consider a network with two input signals comprising the input pattern. There are 2^{2^n} Boolean switching functions for n input signals, so there are 16 ways to recognize the patterns in a 2-D set. If we enumerate the linear equations corresponding to the four input patterns, we get (remembering the 3rd element of 1) the following: $x+y+z=0$, $y+z=0$, $x+z=0$, and $z=0$. These are four planes in Euclidean 3-space, each passing through the origin. The unit sphere is centered about the origin, so all of these planes will intersect it in great circles. Adjust the orientation of the sphere so that the great circles hover around the equator, like a set of ecliptics in quadrature. Now place the south pole of the sphere on a 2-D plane and perforate the sphere at the north pole, projecting it down onto the plane. The great circles are now circles on the plane, each in a different quadrant, and each intersecting all of the others. These circles divide the plane into 14 distinct regions. But there were 16 different ways to recognize a 2 element binary pattern! The network does not have the ability to recognize but 14 of them. Therefore a committee network is functionally deficient. This proof may be generalized by induction to any number of dimensions. QED



A Committee Network of Threshold Logic Units

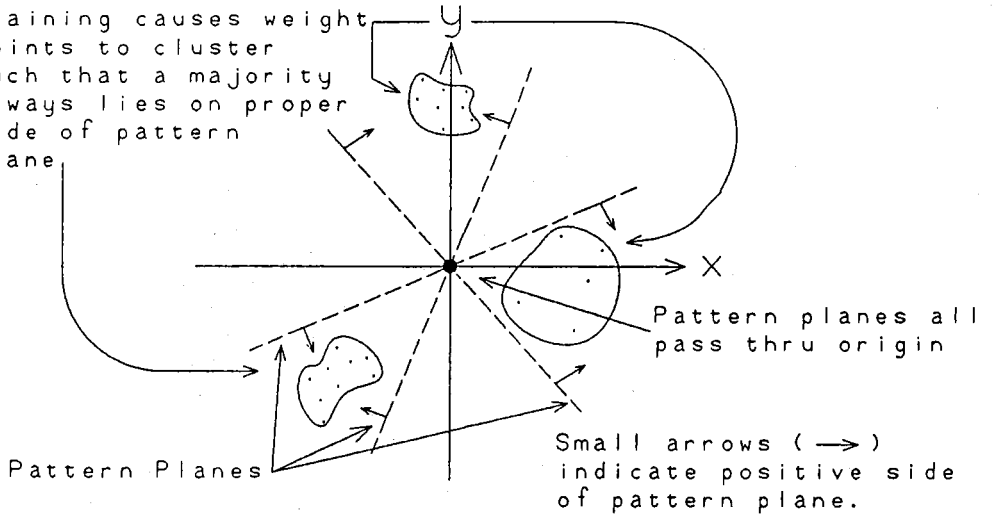


Pattern Planes Intersecting Sphere

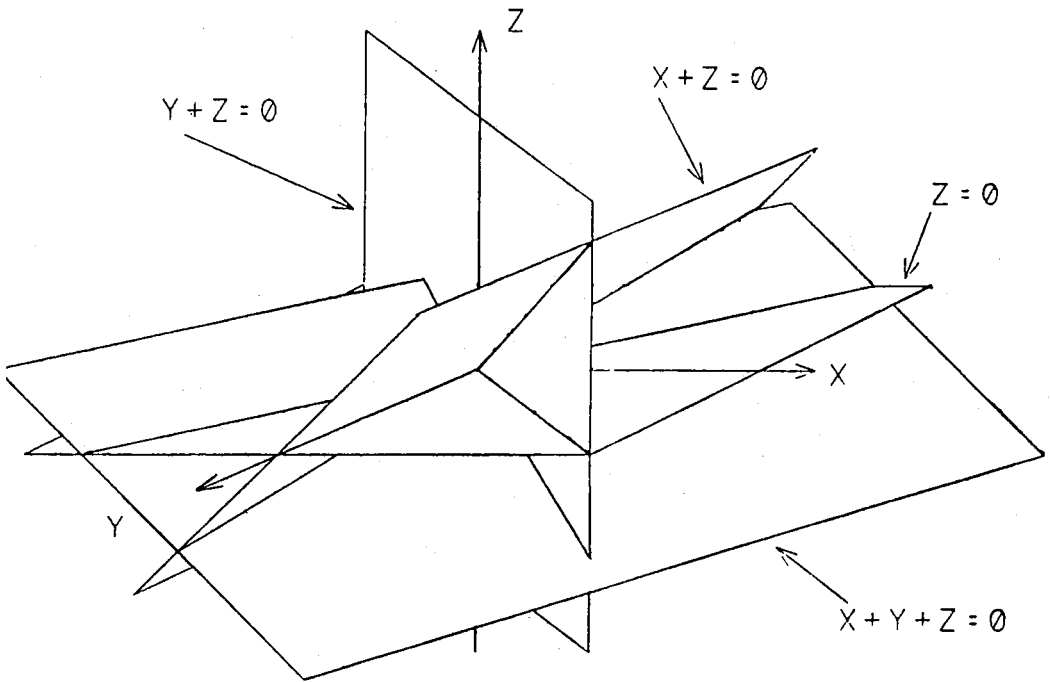


Sphere Mapped to Plane

Training causes weight points to cluster such that a majority always lies on proper side of pattern plane



2-Dimensional Analog for Committee Network



Pattern Planes in 3-Space