Geometry of Perceptrons

Daniel Crespin
Facultad de Ciencias
Universidad Central de Venezuela

Abstract

It is proved that any perceptron neural network is a product of characteristic maps of polyhedra.

1.- **Introduction.** It is known that a product of characteristic maps of linear polyhedra can be realized as a three layer perceptron, see [7]. A more detailed version can be found in [4]. The present paper proves that linear perceptrons are *always* products of characteristic maps of polyhedra. This gives the geometric structure of perceptrons. The proof is elementary and applies to more general perceptron networks as explained below.

In many applications perceptron neural networks are constructed along the following lines. Based on the specific problem at hand an educated guess is made as to the number of layers, and of cells in each layer, necessary for the perceptron network to perform the desired task. The discontinuous activation function typical of perceptrons, known as Heaviside function, is then replaced by a smooth sigmoid. The reason for this replacement is that since the Heaviside function has zero derivative except at the origin (where it is discontinuous) it is a 'gradient blind' function; when approximated by the gradient sensitive sigmoid various gradient optimization procedures, called backpropagation, can be used. Initial weights are then chosen for the various connections. Because the choice is usually made at random, it is often the case that this initial network does not even approximates the performance originally required. Training is then applied. This normally consists in the iterative modification of the network weights, based on data (examples and counterexamples) and aiming at the minimization of a function that gives a measure of how badly a given set of weights performs the task. At the end of the training the smooth sigmoid is replaced back by the Heaviside function.

Many variants of the outlined procedure exist. When training is successful the resulting perceptron network performs, at least approximatedly, as originally demanded. But often the training results inadequate as revealed in poor network performance.

However, once it is established that perceptrons are characteristic maps of polyhedra, an alternative approach is to build directly a polyhedron out of the examples and counterexamples in such a way that it accurately performs the desired task, at least on the data initially provided. This polyhedron can then be realized as a perceptron network and its performance on fresh data can be evaluated. An algorithm to specify these networks has been proposed in [5].

Perceptron units are characteristic maps of linear half spaces in $\mathbf{R}^m$ and polyhedra appear as members of the algebra of sets (a collection closed under intersections an complements) generated by the half spaces. The role of the first layer of a linear perceptron network is to provide a parametrized family of half spaces. The second layer specifies a collection of intersections of half-spaces. These intersections are convex cells. Each output arrow from the third layer selects a collection of convex cells from the second layer. Further layers have the effect of modifying the previous selection of convex cells. The final result is that individual output arrows from the last layer equal the characteristic map of a union of convex cells (defined by the second layer) this union being, by definition, a polyhedron.

The half-spaces defined by the first layer generate an algebra $\mathcal{P}$ of subsets (collection closed under intersections and complements) of euclidean space. All members of $\mathcal{P}$ are polyhedra, in particular the polyhedra determined by the perceptron network. More generally, if Heaviside functions are kept and linear forms f are replaced by quadratic, polynomial, periodic or other kinds of functions g, then the first layer defines generalized half spaces by the condition $h \circ g = 1$. An algebra of subsets of euclidean space is generated by these half spaces (subsets that can be called quadratic, polynomial, periodic or generalized polyhedra), and the arguments below prove also that the corresponding generalized perceptron networks are products of characteristic maps of generalized polyhedra. For notation and basic concepts see [2]-[4].

2.- **Proof of the Structure Theorem.** Let X be a set, $A \subseteq X$, and let $A^{(1)} = A$, $A^{(0)} = X - A$. If $\chi_A : X \to \{0,1\}$ is the characteristic map of A, it has inverse images $\chi^{-1}(b_i) = A^{(b_i)}$, $b_i \in \{0,1\}$ a *bit*. And any map $f : X \to \{0,1\}$ is the characteristic map of a subset, namely, $f = \chi_A$ with $A = f^{-1}(1)$. For n subsets $A_1, \ldots, A_n$ of X the *characteristic indicator* is the product of the characteristic maps $\chi = \chi_{A_1 \cdots A_n} = \chi_{A_1} \times \cdots \times \chi_{A_n} : X \to \{0,1\}^n$, and the former are the sets *indicated* by χ. Furthermore, any map $f = (f_1, \ldots, f_n) : X \to \{0,1\}^n$ is an indicator, namely, the indicator of the sets $A_1 = f_1^{-1}(1), \ldots, A_n = f_n^{-1}(1)$. Consider for binary vectors $b = (b_1, \cdots, b_n) \in \{0,1\}^n$ the intersection $A^{(b)} = A_1^{(b_1)} \cap \cdots \cap A_n^{(b_n)}$. Then $\chi^{-1}(b) = A^{(b)}$ and for subsets $B \subseteq \{0,1\}^n$, $\chi^{-1}(b) = \bigcup\{A^{(b)} | b \in B\}$. It follows that for any $g = (g_1, \ldots, g_p) : \{0,1\}^n \to \{0,1\}^p$ the composition $f = g \circ \chi : X \to \{0,1\}^p$ is the indicator of the sets $P_1 = \chi^{-1}(B_1), \ldots, P_p = \chi^{-1}(B_p)$ where $B_j = g_j^{-1}(1)$, $j = 1, \ldots, p$ and each of the sets indicated by $g \circ \chi$ are finite unions of finite intersections of the A_i's. The structure theorem follows because, on one hand, perceptron layers are characteristic indicators, on the other hand, the first layer is a product of characteristic maps, and finally, the composition of the remaining layers can be replaced by its restriction to $\{0,1\}^n$, say, by $g = (g_1, \ldots, g_p) : \{0,1\}^n \to \{0,1\}^p$ resulting in a function identical to the one defined by the network.

REFERENCES

[1] Crespin, D. Neural Network Formalism.
[2] Crespin, D. Generalized Backpropagation. To appear.
[3] Crespin, D. Geometry of Perceptrons (this present paper).
[4] Crespin, D. Neural Polyhedra .
[5] Crespin, D. Pattern recognition with untrained perceptrons.
[6] Crespin, D. Feature Extraction. To appear.
[7] Lippmann, R. An Introduction to Computing with Neural Nets, IEEE ASSP *Magazine*, April 1987, 4-22.

Daniel Crespin
dcrespin@euler.ciens.ucv.ve
Caracas, December 15, 1995.