Extraction of Low Order Boolean Rules from Trained Neural Networks using a Computer Algebra System

Terence Etchells
School of Computing & Mathematical Sciences
Liverpool John Moores University
Liverpool, UK L3 3AF

Abstract
This paper deals with the extraction of low order and hence meaningful Boolean rules from trained neural networks. Tsukimoto and Morita [1] have developed a scalar algebraic model for classical logic and an algorithm for extracting the nearest Boolean rule from the units of a neural network. We deal with its implementation in the computer algebra system Derive for Windows and discuss the further development of the algorithm to networks whose variables are not necessarily Booleanly 'high' or 'low'.

In essence a Boolean function can be expressed as a multi-linear function which in turn can be expressed as a logic vector, for example, the Boolean expression $x \vee y = x + y - xy$ in the scalar model, which can be written as $xy + x(1-y) + (1-y)x$.
In turn this can be represented as a vector of atoms, where $xy = [1,0,0,0]$ and $x\bar{y} = [1,0,0,0]$ etc. Hence the logic vector for $x \vee y = [1,1,1,0]$, which is of course the truth table for $x \vee y$.

The unit of a neural network is say, $S(w_1x + w_2y + w_3z + b)$ where: *x*, *y* and *z* are inputs; *b* is the bias term; *S*() is the sigmoid function. If $x, y, z \in \{0,1\}$ then the expression
$S(w_1x + w_2y + w_3z + b) = a_1xyz + a_2xy(1-z) + a_3x(1-y)z + a_4x(1-y)(1-z) + \cdots$
is true. If $x, y, z \in [0,1]$ then the above expression is approximately true. Hence the unit of neural network can approximated as a multilinear function, which in turn means we can find the nearest Boolean expression that 'explains' the activation of that unit. In this paper we address the implementation of such an algorithm in Derive for Windows.

A major drawback to this technique is that a neural net with 60 variables leads to logic vectors of size 2^{60}, which is computationally prohibitive. We also present a polynomial algorithm that extracts the low order Boolean expressions, for example orders 1 and 2 in the example below

$$\overbrace{\bar{x}}^{order\,1} \vee \overbrace{xy}^{order\,2} \vee \overbrace{x\bar{y}z \vee y\bar{z}w}^{order\,3} \vee \overbrace{\bar{x}yz\bar{w} \vee vwus}^{order\,4}$$

and its implementation in Derive for Windows.

References

[1] An Algorithm for Extracting Propositions form Trained Neural Networks Using Multilinear Functions, Hirosho Tsukimoto & Chie Morita, Discovery in AI Workshops 1994 , pp103-104.