Stratovan

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

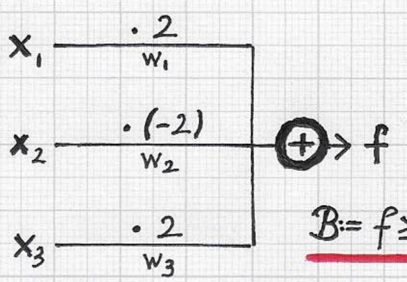
• Laplacian eigenfunctions and neural networks: ... • **On the relationships between Boolean logic, threshold logic units, Boolean operators and expressions, arithmetic operators and expressions, neural networks, learning, training and geometry.** In various places of these notes, it has been pointed out that these topics, areas and concepts are closely related. For example, it has been mentioned and described at a high level that the Boolean operators OR ($\vee$) and AND ($\wedge$) operating on Boolean variables with values TRUE or FALSE have "equivalent" arithmetic operators '$+$' and '$\cdot$' operating on variables with values in $\mathbb{R}$. A relationship can be established as follows: A Boolean variable is discrete and can take on only two possible values; thus, by introducing a threshold, one can view a real variable's value representing TRUE (FALSE) when it is equal to or larger (smaller) than the specified threshold. This view opens the possibility to threshold logic and threshold logic units. Formally, a threshold logic Boolean function computes the truth value ($1 \hat{=}$ TRUE or $0 \hat{=}$ FALSE) for $\langle \mathbb{w}, \mathbb{x} \rangle = \sum_{i=1}^{n} w_i x_i \geq T$, where real weights w_i and real inputs x_i define the weight and input vectors, i.e., $\mathbb{w} = (w_1, \ldots, w_n)^T$ and $\mathbb{x} = (x_1, \ldots, x_n)^T$.

...

Stratovan

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

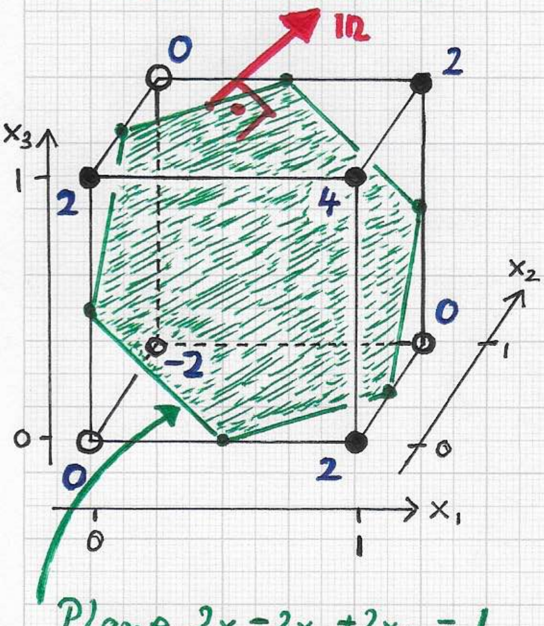
- *Laplacian eigenfunctions and neural networks: ...*



$\mathcal{B} := f \geq 1$

x_1	x_2	x_3	f	$\mathcal{B}$
0	0	0	0	0
1	0	0	2	1
0	1	0	-2	0
1	1	0	0	0
0	0	1	2	1
1	0	1	4	1
0	1	1	0	0
1	1	1	2	1

$f = 2x_1 - 2x_2 + 2x_3$



Plane $2x_1 - 2x_2 + 2x_3 = 1$ clipped against the unit cube $[0,1]^3$. The plane is the DECISION BOUNDARY. The vector $\mathbb{n}$ is the outward normal.

We consider a simple example to point out the relationships between algebra, geometry and logic. We use – initially – binary input variables x_1, x_2 and x_3 that can have values 0 or 1. Further, we use the weight vector $\mathbb{w} = (2,-2,2)^T$ and the threshold $T = 1$. The left figure shows the threshold logic unit (top) and the associated input-and-output table for all $2^3 = 8$ binary input vectors. The values of $f = \langle \mathbb{w}, \mathbb{x} \rangle = 2x_1 - 2x_2 + 2x_3$ are compared with the threshold value $T = 1$, and the comparison result defines the binary Boolean truth value of $\mathcal{B}$. The value of $\mathcal{B}$ is 1 (TRUE) if $f \geq 1$; otherwise it is 0 (FALSE).

The equation $2x_1 - 2x_2 + 2x_3 = 1$ is an implicit plane equation that separates 3D space into a positive and negative half-space, i.e., the half-spaces $\langle \mathbb{w}, \mathbb{x} \rangle > 0$ and $\langle \mathbb{w}, \mathbb{x} \rangle < 0$, see left figure (bottom).

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

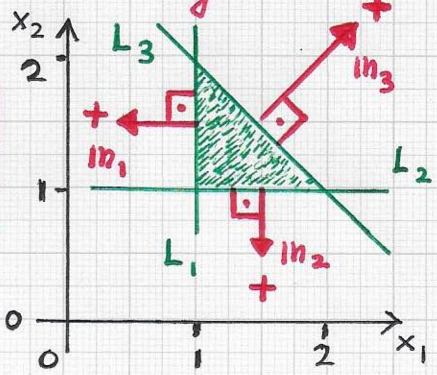
• Laplacian eigenfunctions and neural networks:... Thus, the implicitly defined (hyper-) plane $\langle \mathbf{w}, \mathbf{x} \rangle = \text{const}$ determines the Boolean output, **1** or **0**. The plane itself is the so-called **DECISION BOUNDARY**. Conventionally, points in this plane are defined as TRUE points. The figure on the previous page (bottom) use '●' to indicate cube vertices with values above $T=1$ and 'o' to indicate cube vertices with values below $T=1$. The OUTWARD normal vector of the plane is given by the gradient (vector) of the function $f(x_1, x_2, x_3) = 2x_1 - 2x_2 + 2x_3 - 1$, i.e., the tuple $(2, -2, 2)$. The gradient (vector) "points into the positive half-space." This vector is called $\mathbf{n}$ in the figure. Further, a point (x_1, x_2, x_3) with REAL-VALUED coordinates is a TRUE (FALSE) point if $\langle \mathbf{w}, \mathbf{x} \rangle \geq T$ ($\langle \mathbf{w}, \mathbf{x} \rangle < T$). Thus, we can apply the threshold logic unit to real-valued input variables x_i – as desired.

The binary, standard Boolean logic result table presented on the previous page allows one to define the output B as a function $B(x_1, x_2, x_3)$ – with $x_1, x_2, x_3, B \in \{0, 1\}$ – in many equivalent algebraic expressions, e.g., $B = (x_1 \wedge \bar{x}_2) \vee (x_1 \wedge x_3) \vee (\bar{x}_2 \wedge x_3)$.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... An optimal, i.e., "simplest," form for the algebraic representation of $\mathbb{B}$ can be found via the **Karnaugh-Veitch diagram (K-map)**, for low-dimensional input vectors $\mathbf{x}$. Of course, the desire to represent $\mathbb{B}$ simply is driven by the need of computational efficiency. **We now "see" the relationships between and "equivalence" of algebra, geometry and logic — and will be able to utilize these relationships for material classification purposes and neural network design.**



MULTIPLE DECISION BOUNDARIES. The left figure illustrates the use of multiple line decision boundaries to define a region in the plane, i.e., the shaded triangle. The triangle is the intersection of three negative half-spaces defined by three line equations, L_1, L_2 and L_3. The three equations are L_1: $f_1(x_1,x_2) = -x_1+1=0$; L_2: $f_2(x_1,x_2) = -x_2+1=0$ and L_3: $f_3(x_1,x_2) = x_1+x_2-3=0$. The gradients (outward normal vectors) of these line functions are $\mathbf{n}_1 = (-1,0)^T$, $\mathbf{n}_2 = (0,-1)^T$ and $\mathbf{n}_3 = (1,1)^T$. These vectors (not normalized here) point into the lines' **positive (+) half-spaces**.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

It is possible to use this definition of a (convex) region via the intersection of (convex) half-spaces in the context of classification. For example, if the points $(x_1, x_2)^T$ inside the triangle represented a "class," then the logical expression

$$f_1 \le 0 \;\wedge\; f_2 \le 0 \;\wedge\; f_3 \le 0$$

would have to yield the value TRUE (1) for a given point $(x_1, x_2)^T$ to indicate that it lies in the interior or on the boundary of the triangle and therefore represents the "class."

In the general multi-dimensional case, (hyper-) half-spaces can be intersected to define finite, closed, compact, convex regions. The faces/facets of such convex regions are defined by (properly oriented) hyper-planes/linear equations. More complicated NON-CONVEX regions can be constructed as UNIONS of convex regions. Further, we must keep in mind that one can approximate the boundary of a complicated region, bounded by liner (hyper-) faces/facets, with arbitrarily small error — by increasing the number of linear boundary elements (linear (hyper-)plane equations) to the necessary number.

...