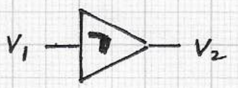
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

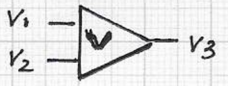
• Laplacian eigenfunctions and neural networks:...

Symbols and notation used for Boolean operators, gates and circuits:



$V_2 := \neg V_1$

$V_3 := V_1 \wedge V_2$

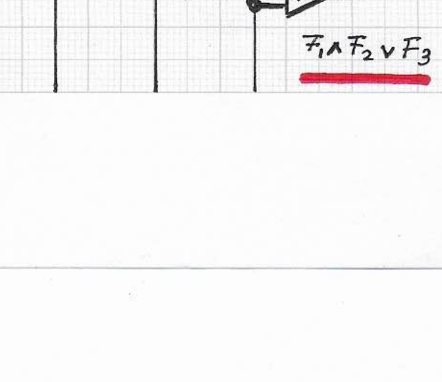


$V_3 := V_1 \vee V_2$

Simplified illustrations for the three basic Boolean operators.

(Ultimately, in the context of fuzzy logic, the purely binary decision-making will be replaced by probabilistic decision-making and generalized versions of the basic Boolean operators.)

Viewing the values of three univariate decider functions as binary truth variable values, one can illustrate, for example, the computation of $F_1 \wedge F_2 \vee F_3$ as follows:



•**Note.** Since Boolean algebra and its operators are important for this discussion, we briefly summarize the essential rules. The three basic operators are written as $\neg$ (NOT), $\wedge$ (AND) and $\vee$ (OR); V_1, V_2 and V_3 are Boolean variables. These are the fundamental rules:

$$V_1 \vee V_2 = V_2 \vee V_1 \quad \text{C}$$
$$V_1 \wedge V_2 = V_2 \wedge V_1 \quad \text{C}$$
$$(V_1 \vee V_2) \vee V_3 = V_1 \vee (V_2 \vee V_3) \quad \text{A}$$
$$(V_1 \wedge V_2) \wedge V_3 = V_1 \wedge (V_2 \wedge V_3) \quad \text{A}$$
$$V_1 \wedge (V_2 \vee V_3) = (V_1 \wedge V_2) \vee (V_1 \wedge V_3) \quad \text{D}$$
$$V_1 \vee V_2 \wedge V_3 = V_1 \vee (V_2 \wedge V_3) \quad \text{P}$$

!

These examples capture commutative (C), associative (A) and distributive (D) rules; the $\wedge$ operator has higher precedence (P) than the $\vee$ operator. The order of decreasing precedence of all three basic operators is: $\neg$, then $\wedge$, then $\vee$. Keeping these rules in mind and using the symbols and notation illustrated in the figures (left), one can devise the Boolean circuits for the six cases (previous page):

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

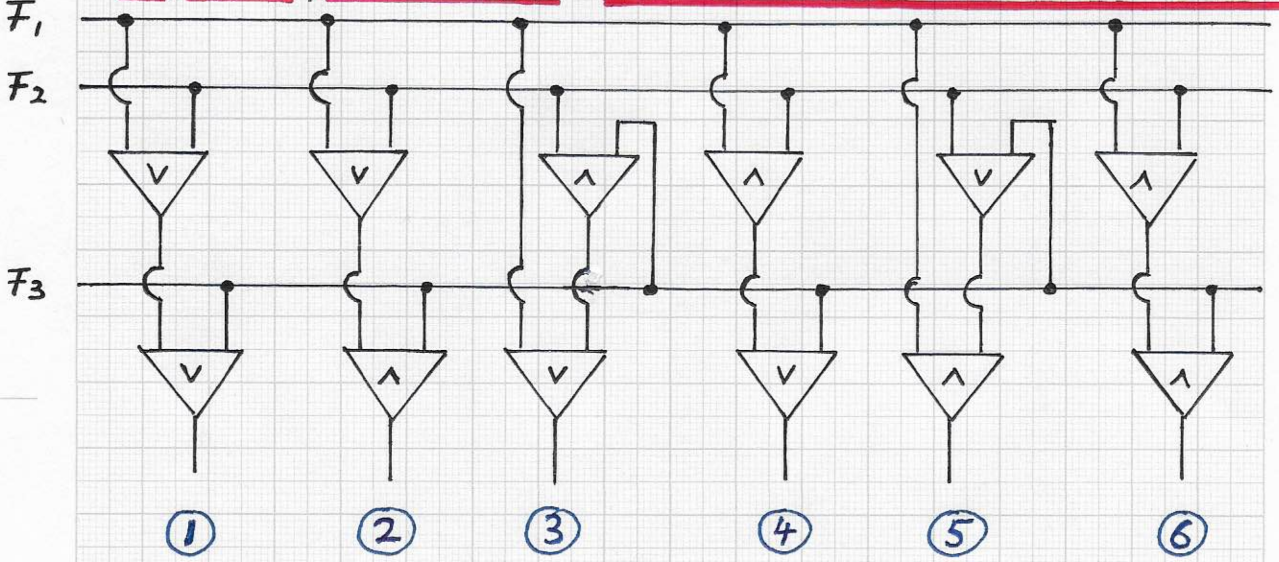
• Laplacian eigenfunctions and neural networks: ...

(When using "binary Boolean logic," we view low [high] values of decider functions as FALSE [TRUE].)

Input: (F_1, F_2, F_3) value triple

Output: 6-tuple for cases 1, ..., 6

CIRCUIT ARCHITECTURE FOR 3 SCALES:



This architecture produces the 48 possible output values show in the table on p. 5 from 6/20/2022.

This architecture summarizes succintly the Boolean logic computations when three univariate decider functions are used, e.g., functions F_1, F_2 and F_3 for the coarsest, lowest-frequency scales 1, 2 and 3. Eight binary input triples are possible; the combinatorically possible ways to "combine" triple values via the $\wedge$ and $\vee$ operators lead to the shown six cases 1, ..., 6.

FUZZY LOGIC!

Generally, different weights must be associated with different-scale univariate decider functions. For example, low-frequency functions F_1 and F_2 should carry significantly more weight than functions F_{H-1} and F_H.

Alternatively, one might know that classification-relevant information is captured in specific scales — which should thus have higher weights.

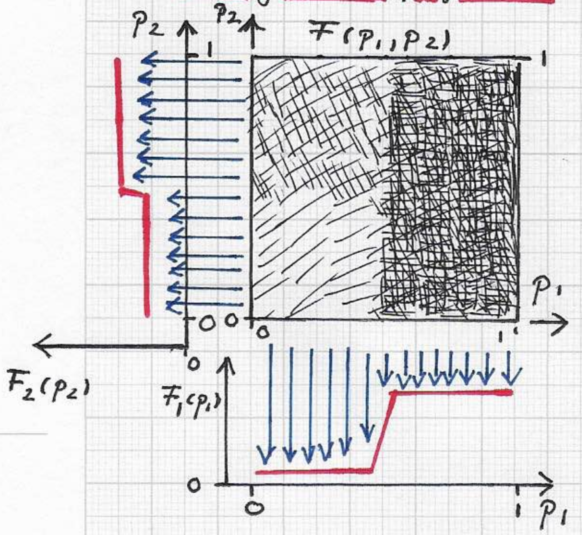
...

Stratovan

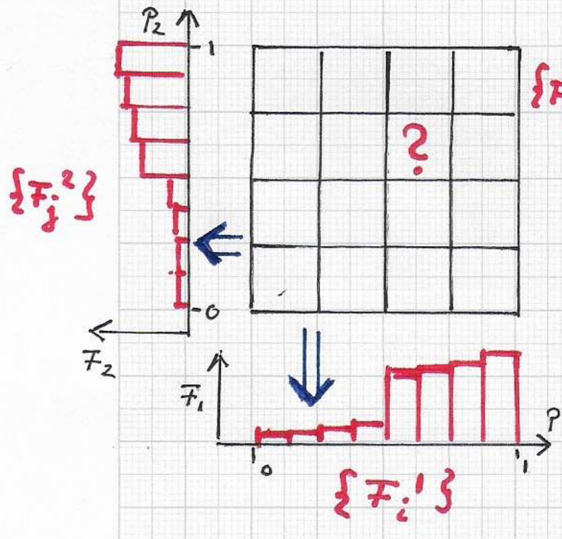
■ OBJECT AND MATERIAL EIGENFUNCTIONS — Cont'd.

• Laplacian eigenfunctions and neural networks: ...

Interpretation of univariate decider functions as integrations, projections:



Abstract view of LINE INTEGRALS defining the function values of F_1 and F_2.



Discrete solution method: Use the two finite sets of line integral values F_i^1 and F_j^2 to optimally compute values $F_{I,J}$.

The overall decider function, e.g., $F(p_1, p_2)$ in the two-scale case, is not known; known, given are only univariate decider functions $F_1(p_1)$ and $F_2(p_2)$ — defined by an expert or via training. **Thus, one can view the construction of F as a FUNCTION RECONSTRUCTION problem: Use F_1 and F_2 to compute a "best reconstruction" — subject to employing a specific mathematical model for F and corresponding numerical best approximation methods.** The figure (left) illustrates this viewpoint. Ignoring specific numerical values and value constraints, the values of $F_1(p_1)$ and $F_2(p_2)$ are defined by integrals:

$$\int_{p_2=0}^{1} F(p_1, p_2)\, dp_2 = F_1(p_1) \quad , p_1 \in [0,1]$$

$$\int_{p_1=0}^{1} F(p_1, p_2)\, dp_1 = F_2(p_2) \quad , p_2 \in [0,1]$$

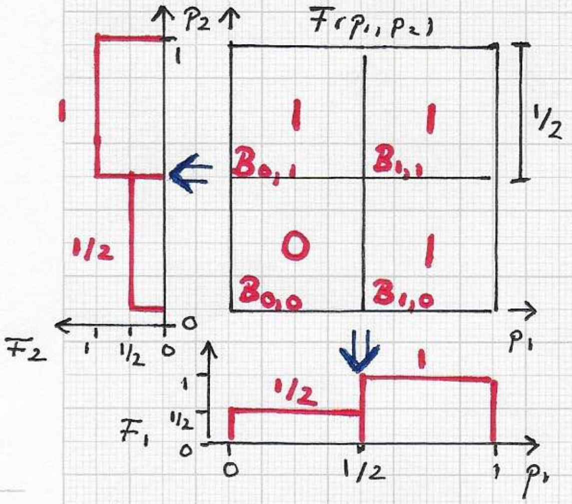
One can discretize the domains of F_1, F_2 and F and devise a numerical solution approach.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

Simple, ideal example:



Principle of (re-) construction: Synthetic, ideal function F on grid of resolution 2-by-2 is "projected." The 4 projected values of F when integrated via summing F-values in p_1- and p_2-directions are used to determine the unknown coefficient values for the box basis functions $B_{I,J}$, $I,J = 0...1$.

The synthetic function is

$$F(p_1, p_2) = 0\,B_{0,0} + 1\,B_{1,0} + 1\,B_{0,1} + 1\,B_{1,1}.$$

F's projection values onto the p_1-axis are

$\frac{1}{2}\cdot 0 + \frac{1}{2}\cdot 1 = \frac{1}{2}$ and
$\frac{1}{2}\cdot 1 + \frac{1}{2}\cdot 1 = 1$.

Analogously, one obtains the projection values $\frac{1}{2}$ and 1 when projecting F onto the p_2-axis.

We consider a very simple example, see figure (Left):

- The model used to represent F is a constant spline, i.e., F's domain is a uniform, equidistant grid of resolution 2-by-2.
- Each box basis function $B_{I,J}$ is associated with grid cell (I,J) – where the basis function has the value 1; it has the value 0 in all other cells.
- The representation of F is
$$F(p_1,p_2) = \sum_{J=0}^{1} \sum_{I=0}^{1} c_{I,J}\, B_{I,J}.$$
- In the example, a function F is shown that is represented error-free with coefficients $c_{0,0} = 0$ and $c_{1,0} = c_{0,1} = c_{1,1} = 1$.
- This function F is "projected" onto the p_1- and p_2-axis, where the interval $[0,1]$ is discretized with resolution 2.
- GIVEN THE DISCRETE VERSIONS OF F_1 AND F_2, ONE WANTS TO (RE-) CONSTRUCT F.

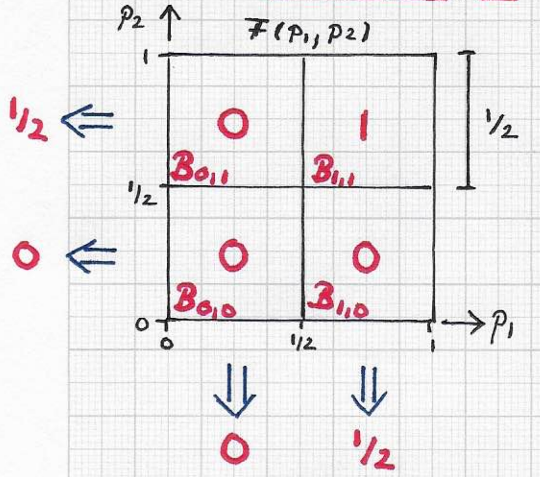
...

Stratovan

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

Second ideal example



The synthetic function is

$$F(p_1,p_2) = 0\,B_{0,0} + 0\,B_{1,0} + 0\,B_{0,1} + 1\,B_{1,1}\,.$$

The projection values are 0 and 1/2 (p_1-axis) and 0 and 1/2 (p_2-axis). The linear system associated with these projection values is

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ c_{1,0} \\ c_{1,1} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

The symbolic, parametric solution in this case is $c_{0,0} = c_{1,1} - 1$; $c_{0,1} = 1 - c_{1,1}$; $c_{1,0} = 1 - c_{1,1}$.

The specific behavior of F_1 and F_2 imply that

$$c_{0,0} = 0$$

$\Rightarrow c_{1,1} = 1;\ c_{0,1} = 0;\ c_{1,0} = 0$.

The reconstruction is perfect.

• In the (re-)construction case, we know the projected values and must calculate – in an optimal way – the unknown values of the coefficients $c_{I,J}$.

• In this example, one obtains the following linear equation system:

$$\begin{bmatrix} \frac{1}{2} c_{0,0} + \frac{1}{2} c_{0,1} \\ \frac{1}{2} c_{1,0} + \frac{1}{2} c_{1,1} \\ \frac{1}{2} c_{0,0} + \frac{1}{2} c_{1,0} \\ \frac{1}{2} c_{0,1} + \frac{1}{2} c_{1,1} \end{bmatrix} \begin{matrix} = \\ = \\ = \\ = \end{matrix} \begin{bmatrix} \frac{1}{2} \\ 1 \\ \frac{1}{2} \\ 1 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ c_{1,0} \\ c_{1,1} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \end{bmatrix}$$

This system is not of full rank. For example, one parametric solution is $c_{0,0} = c_{1,1} - 1$, $c_{0,1} = 2 - c_{1,1}$ and $c_{1,0} = 2 - c_{1,1}$. Considering the behavior of $F_1(p_1)$ and $F_2(p_2)$ – i.e., $F_1(p_1) = 1$ for $p_1 \in [1/2, 1]$ and $F_2(p_2) = 1$ for $p_2 \in [1/2, 1]$ – the value for $c_{1,1}$ must be 1. $\Rightarrow c_{0,0} = 0;\ c_{0,1} = 1;\ c_{1,0} = 1$.

Thus, F has been reconstructed perfectly.

...