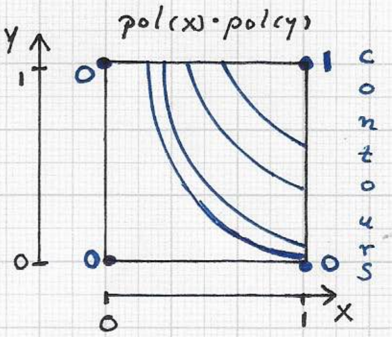
OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks: ...

Examples of R-function 'combinations':



'Combination' of the two Bernstein-Bézier polynomials (from previous page) via multiplication. The figure illustrates the qualitative behavior of the product of the two R-functions:

$$R_1, R_2 = R_1 \circledast R_2 = (3x^2-2x^3)\cdot(3y-3y^2+y^3)$$

- Using the 3D case (H=3) as an example, these are the 'combinations' for all three "RAMP" R-functions:

$R_1 \circledast 0 = R_2 \circledast 0 = R_3 \circledast 0 = \mathbf{0}$;

$R_1 \circledast 1 = R_1$, $R_2 \circledast 1 = R_2$, $R_3 \circledast 1 = R_3$;

$R_1 \circledast R_2$, $R_1 \circledast R_3$, $R_2 \circledast R_3$;

$R_1 \circledast R_2 \circledast R_3$

$\Rightarrow 1+3+3+1 = 8 = 2^H$ distinct 'combinations'

The two simple examples of possible R-function 'combinations' provided on the previous and this page show that the resulting bivariate decider functions exhibit complex isoline / contour behavior. Therefore, one must carefully balance the two primary driving objectives: (i) use of R-functions that have sufficient degrees of freedom to model the individual decider functions $F_i(p_i)$ properly for the intervals (p_1^i, p_2^i); (ii) use of R-functions that can be defined and represented simply, can be evaluated efficiently, can be 'combined' via simple and few additive and/or multiplicative operators, and can be elegantly optimized for for final classification performance. CONVEX, WEIGHTED LINEAR COMBINATIONS ARE GOOD CANDIDATES: $\sum_{i=1}^{H} w_i F_i(p_i)$.

...

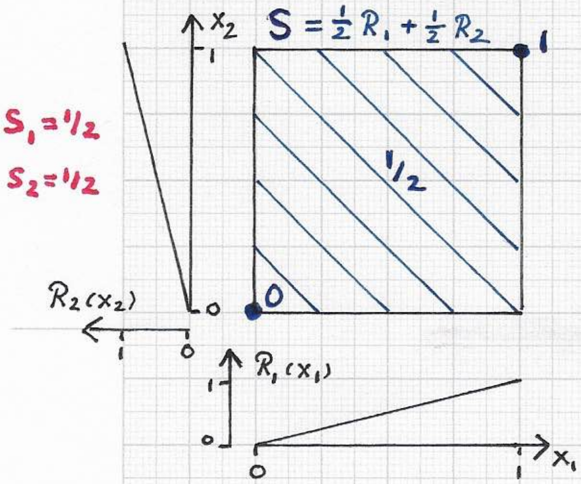
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

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

Example of convex combinations of the "RAMP" part of two univariate decider functions in the normalized parameter domain:

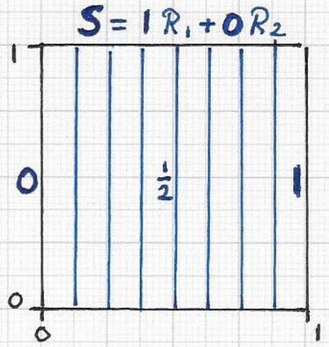
$s_1 = 1/2$

$s_2 = 1/2$



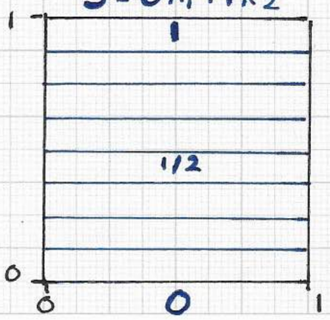
$s_1 = 1$

$s_2 = 0$



$s_1 = 0$

$s_2 = 1$



The shown contours illustrate the substantial impact of weights s_i.

In principle, when faced with the problem of having to define the decider function $F(p_1, \ldots, p_H)$ for its H-dimensional domain cuboid – the unit H-dimensional hyper-cube $[0,1]^H$ – one can 'combine' the individual univariate decider functions $F_i(p_i)$, $i = 1 \ldots H$, in an integrated ADDITIVE–MULTIPLICATIVE manner. For example, one could consider a combination of a weighted SUM of functions $F_i(p_i)$ and a weighted PRODUCT of functions $F_i(p_i)$. The weighted SUM (S) could be defined as a CONVEX COMBINATION:

$$S = \Sigma_{i=1}^{H} s_i F_i(p_i), \quad s_i \geq 0 \wedge \sum_i s_i = 1.$$

The weighted PRODUCT (P) could be defined analogously:

$$P = \Pi_{i=1}^{H} t_i F_i(p_i), \quad t_i > 0 \wedge \prod_i t_i = 1. \oplus$$

A final combination could be defined as

$$C = w S + (1-w) P, \quad 0 \leq w \leq 1.$$

⊕ t_i-values are constant values
$\Rightarrow P = \Pi_{i=1}^{H} F_i(p_i)$.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

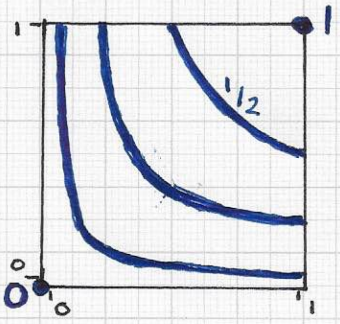
• Laplacian eigenfunctions and neural networks: ...

Examples of weighted PRODUCT (P) functions using varying weight functions $t_1(x_1)$ and $t_2(x_2)$ of the "RAMP" part of the univariate decider functions in the normalized domain:

$R_1(x_1) = x_1$; $R_2(x_2) = x_2$

$t_1(x_1) = 1$; $t_2(x_2) = 1$

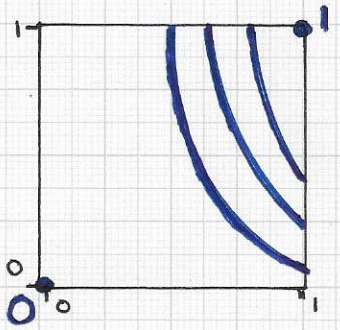
$\Rightarrow P = 1x_1 \cdot 1x_2 = x_1 x_2$



$R_1(x_1) = x_1$; $R_2(x_2) = x_2$

$t_1(x_1) = x_1^3$; $t_2(x_2) = 1$

$\Rightarrow P = x_1^3 x_1 \cdot 1 x_2 = x_1^4 x_2$



Sketches of contours illustrate the impact of weight functions $t_i(x_i)$.

• **Note.** By defining t_i to be a non-constant WEIGHT FUNCTION – i.e., by defining t_i as $t_i = t_i(p_i)$ – the resulting values of P will indeed be different, depending on the choice made for $t_i = t_i(p_i)$. In this case, **the weight functions $t_i(p_i)$ must satisfy the conditions**

i) $t_i(p_i) > 0$, $p_i \in [0,1]$, $i = 1 \dots H$;

ii) $\prod_{i=1}^{H} t_i(p_i) \leq 1$, $p_i \in [0,1]$;

iii) $\prod_{i=1}^{H} t_i(p_i) \Big|_{(p_1,\dots,p_H)=(1,\dots,1)} = 1$;

iv) $t_i(p_i) \leq 1$, $p_i \in [0,1]$, $i = 1 \dots H$.

(Of course, it is possible to add more restrictions and conditions to this list or to be less restrictive.)

• **Note.** Using the weights $s_1 = s_2 = \frac{1}{2}$ and $t_1 = t_2 = 1$, one obtains:

$$S(x_1,x_2) = \tfrac{1}{2}x_1 + \tfrac{1}{2}x_2 \ , \quad P(x_1,x_2) = x_1 x_2 \ ,$$

$$C(x_1,x_2) = \tfrac{w}{2}x_1 + \tfrac{w}{2}x_2 + (1-w)x_1 x_2 = \textstyle\sum_{j=0}^{1}\sum_{i=0}^{1} c_{i,j}\, x_1^i x_2^j \ ,$$

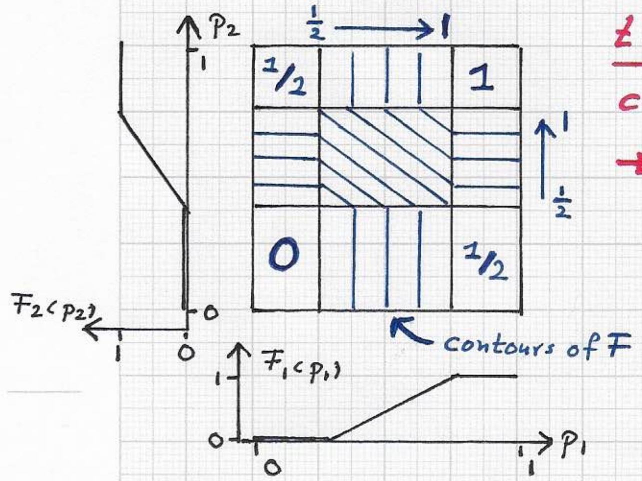
where $c_{0,0} = 0$, $c_{1,0} = c_{0,1} = \frac{w}{2}$, $c_{1,1} = (1-w)$. Thus, $C(x_1,x_2)$ emphasizes linear or quadratic behavior when the value of w is close to 1 or close to 0, respectively.

...

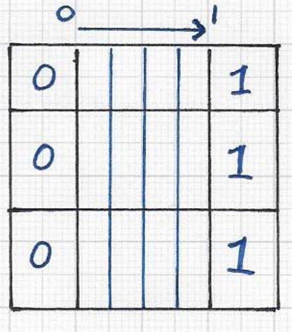
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

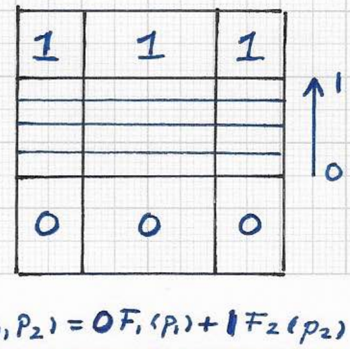
Examples of simple convex combinations of two univariate decider functions $F_1(p_1)$ and $F_2(p_2)$:



$F(p_1,p_2) = \frac{1}{2} F_1(p_1) + \frac{1}{2} F_2(p_2)$



$F(p_1,p_2) = 1 F_1(p_1) + 0 F_2(p_2)$



$F(p_1,p_2) = 0 F_1(p_1) + 1 F_2(p_2)$

It is important to remind oneself that practical aspects should drive the design of the individual decider functions $F_i(p_i)$ and overall decider function $F(p_1,...,p_H)$:

→ The "RAMP" parts of all functions $F_i(p_i)$ must be as simple as possible and, at the same time, yet flexible and complex to satisfy the required classification performance – considering computational time complexity and detection capability.

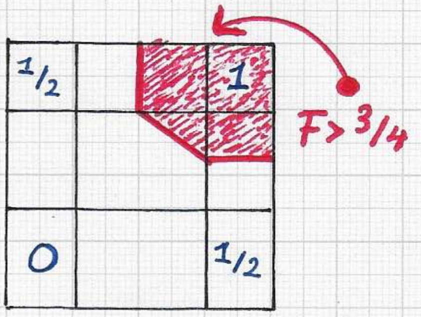
→ The 'combination' of all functions $F_i(p_i)$ used to define $F(p_1,...,p_H)$ must also satisfy the same requirements, where 'combining' "RAMP" functions over $F(p)$'s domain cuboids is crucial; e.g., the optimization of 'combination' weights must be effective and efficient. ...

Bernd Hamann
Stratovan

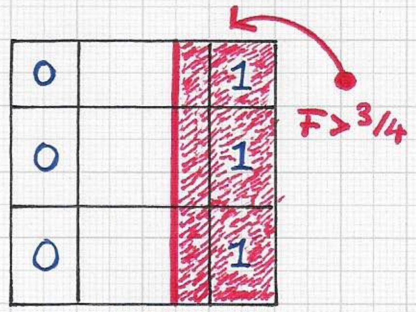
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

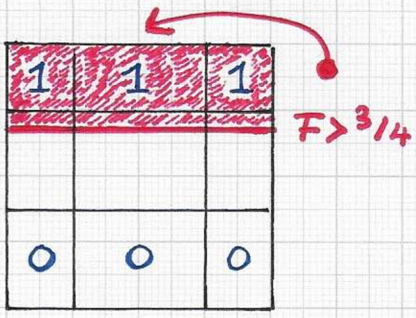
Example of regions in (p_1, p_2)-domain defining the classification regions for a specific threshold:



$F = \frac{1}{2} F_1 + \frac{1}{2} F_2$



$F = F_1$



$F = F_2$

The resulting classification regions (shaded) are significantly different, depending on the weights chosen for the convex combinations.

We now consider a convex combination of only two univariate decider functions $F_1(p_1)$ and $F_2(p_2)$, to understand the influence of the chosen weights on the final overall decider function $F(p_1, p_2)$. The convex combination we consider is

$$F(p_1, p_2) = w_1 F_1(p_1) + w_2 F_2(p_2).$$

The function F is used to determine whether the value of a given tuple (p_1, p_2) **indicates** that the associated material – represented as an image segment – belongs to a specific material class or not. **The two functions F_1 and F_2 are viewed as being TRUE in the regions in their [0, 1] domain where they have values 0 and 1. Based on only one parameter – p_1 and p_2, respectively – they decide whether the material being analyzed is (1) or is not (0) of a specific class. The function F is used to make the decision via an OPTIMAL CONVEX COMBINATION of F_1 and F_2.**

...