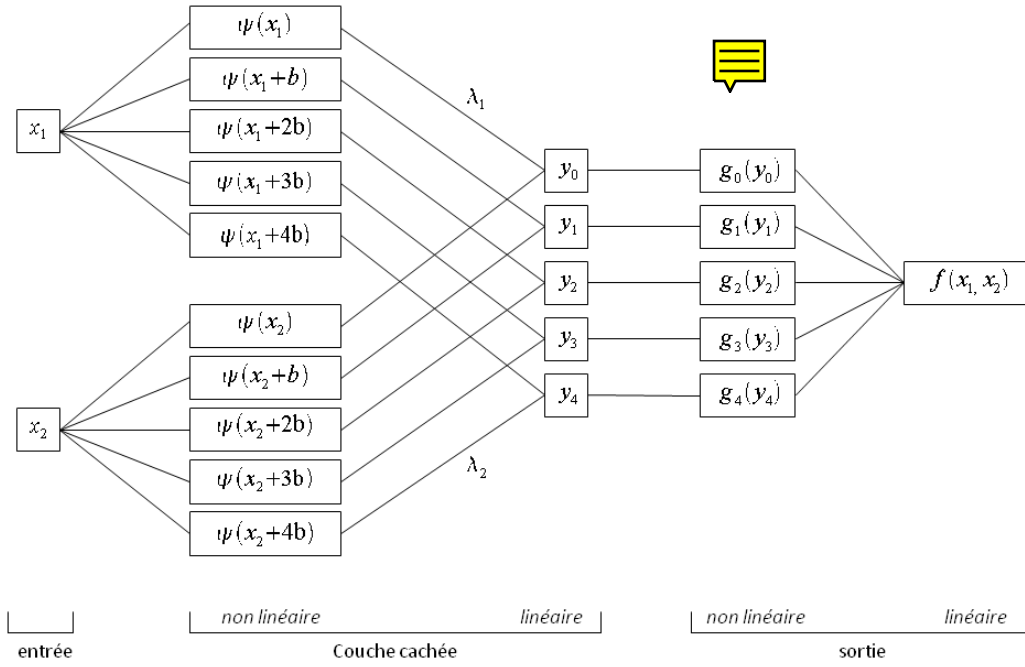


# Chapitre 2

## Implémentations

Dans le chapitre précédent, nous avons présenté le théorème de superposition de Kolmogorov et les différentes contributions proposées depuis sa présentation en 1957, notamment pour la construction des fonctions monodimensionnelles. Dans notre étude de la décomposition d'images par le théorème de superposition, nous avons tout d'abord implémenté l'algorithme proposé par Sprecher. Nous nous sommes intéressés en premier à cet algorithme parce que la décomposition proposée est fidèle au théorème, et son énoncé et son implémentation sont particulièrement accessibles. Nous avons ensuite implémenté l'algorithme d'Igelnik et Parikh pour pouvoir bénéficier d'une autre décomposition, plus ajustable, et disposant d'une représentation analytique des fonctions internes et externes. Très rapidement, nous avons observé que l'algorithme de Sprecher ne permettait pas une aussi grande flexibilité alors que celui d'Igelnik, notamment grâce à la représentation continue des fonctions internes et externes, ouvrait la porte à de plus amples adaptations. C'est la raison pour laquelle nos contributions présentées dans les chapitres 3 et 4 s'appuient sur des modifications de l'algorithme d'Igelnik. Nous présentons dans ce chapitre ces deux algorithmes et leur application à la décomposition d'images en niveaux de gris.



**Figure 2.1** – Illustration de l’analogie entre le TSK et un réseau de neurones à une couche cachée.

## 2.1 L’algorithme de Sprecher

### 2.1.1 Introduction

L’algorithme de Sprecher s’appuie sur la formulation du théorème de Kolmogorov sous la forme suivante :

$$\begin{cases} f(x_1, \dots, x_d) = \sum_{n=0}^{2^d} g_n(\xi(x_1 + bn, \dots, x_d + bn)) \\ \xi(x_1 + bn, \dots, x_d + bn) = \sum_{i=1}^d \lambda_i \psi(x_i + bn) \end{cases} \quad (2.1)$$

Cette formulation peut être interprétée comme une représentation de la fonction multidimensionnelle par un réseau de neurones à une couche cachée comme l’a montré Hecht-Nielsen et comme illustré en Figure 2.1.

Sprecher a proposé un algorithme exact, à une précision donnée, pour la construction des fonctions internes et externes dans [Sprecher, 1996] et [Sprecher, 1997], respectivement. Comme la fonction  $\psi$  initialement définie par Sprecher pour construire  $\xi$  est discontinue pour certaines valeurs d'entrée, nous utilisons la fonction  $\psi$  proposée par Braun et Griebel dans [Braun and Griebel, 2009] à la continuité et monotonie démontrées, qui corrige les problèmes de discontinuité aux extrémités des intervalles de la fonction  $\psi$  introduite par Sprecher. Le reste de l'algorithme de Sprecher n'est pas modifié, *i.e.* les fonctions monodimensionnelles internes et externes restent évaluées à une précision donnée pour n'importe quelle coordonnée du domaine de définition.

### Définition 1 (Notations)

- $d$  est la dimension,  $d \geq 2$ .
- $m$  est le nombre de couches de fonctions,  $m \geq 2d$ .
- $\gamma$  est la base des variables  $x_i$ ,  $\gamma \geq m + 2$ .
- $b = \frac{1}{\gamma(\gamma-1)}$  est la translation entre deux couches de pavage.
- $\lambda_1 = 1$  et pour  $2 \leq i \leq d$ ,  $\lambda_i = \sum_{r=1}^{\infty} \frac{1}{\gamma^{(i-1)(d^r-1)/(d-1)}}$  sont les coefficients de la combinaison linéaire qui sert d'argument aux fonctions  $g_n$ .

### 2.1.2 Algorithme

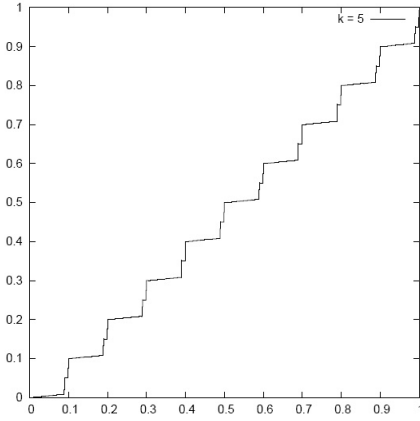
Les travaux de Sprecher portent sur la calculabilité de la fonction  $\xi$  et la définition de  $\psi$ . Plus précisément, la construction de  $\psi$  et la structure de l'algorithme sont basées sur la décomposition des nombre réels dans la base  $\gamma$ ; *i.e.* on peut écrire tout nombre décimal (noté  $d_k$ ) entre  $[0, 1]$  à  $k$  décimales sous la forme :

$$d_k = \sum_{r=1}^k i_r \gamma^{-r}, \text{ et } d_k^n = d_k + n \sum_{r=2}^k \gamma^{-r} \quad (2.2)$$

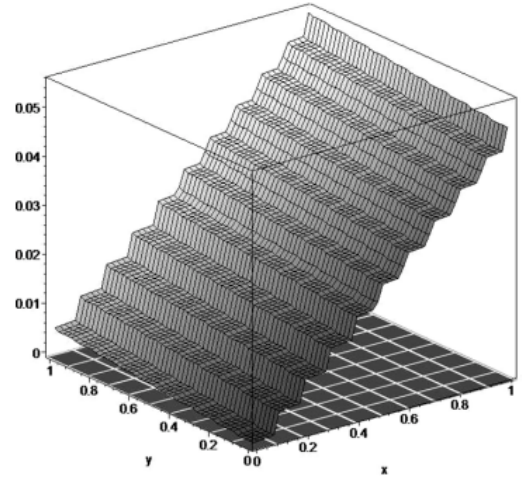
En utilisant les  $d_k$  définis en équation 2.2, Braun *et al.* définissent la fonction  $\psi$  par :

$$\psi_k(d_k) = \begin{cases} d_k & \text{pour } k = 1, \\ \psi_{k-1}(d_k - \frac{i_k}{\gamma^k}) + \frac{i_k}{\gamma^{\frac{d^k-1}{d-1}}} & \text{pour } k > 1 \text{ et } i_k < \gamma - 1, \\ \frac{1}{2}(\psi_k(d_k - \frac{1}{\gamma^k}) + \psi_{k-1}(d_k + \frac{1}{\gamma^k})) & \text{pour } k > 1 \text{ et } i_k = \gamma - 1. \end{cases} \quad (2.3)$$

La figure 2.2(a) représente le graphe de la fonction  $\psi$  sur l'intervalle  $[0, 1]$ . La fonction  $\xi$  est calculée par combinaison linéaire des réels  $\lambda_i$  et de la fonction  $\psi$  appliquée à chaque composante  $x_i$  de la valeur d'entrée. La figure 2.2(b) représente la fonction  $\xi$  sur l'espace  $[0, 1]^2$ .



(a)

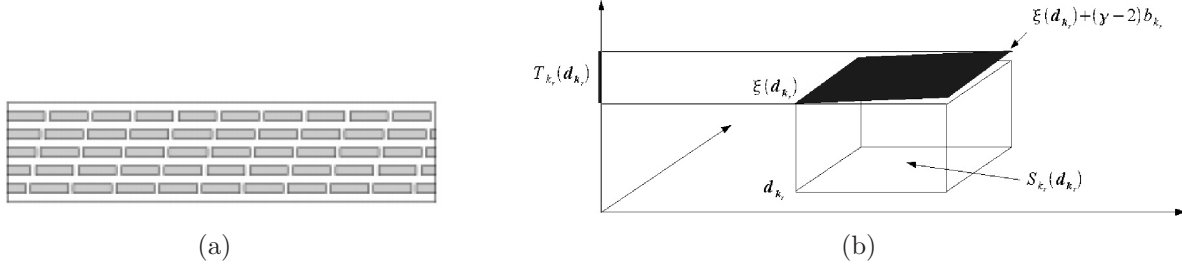


(b)

**Figure 2.2** – (a) Graphe de la fonction  $\psi$  pour la base  $\gamma = 10$ , d'après [Braun and Griebel, 2009]. (b) Fonction de hashage  $\xi$  pour  $d = 2$  et  $\gamma = 10$ , d'après [Brattka, 2004].

Sprecher a montré que chacun des intervalles disjoints  $I$  était associé à un unique intervalle  $\psi(I)$ . Cette propriété de séparation est utilisée pour obtenir des intervalles définissant un pavage disjoint de  $[0, 1]$ . Cette méthode est étendue à un espace  $d$ -dimensionnel en effectuant le produit cartésien des intervalles  $I$ . Afin de couvrir tout l'espace, le pavage est translaté plusieurs fois d'une constante  $b$ , ce qui produit les différentes couches finales du pavage. On obtient ainsi  $2d + 1$  couches, le pavage original constitué par des hypercubes disjoints ayant des images disjointes par la combinaison linéaire des fonctions  $\psi$ , et  $2d$

pavages translatis d'une constante  $b$  le long de chaque dimension. La figure 2.3(a) représente une coupe du pavage complet pour un espace  $2D$  : on voit les  $2d + 1 = 5$  pavages différents, décalés par  $b$ , qui se superposent.



**Figure 2.3** – (a) Vue en coupe d'un pavage dans le cas  $2D$  avec  $\gamma = 10$  (5 couches différentes), d'après [Brattka, 2004]. (b) La fonction  $\xi$  met en correspondance chaque pavé avec un intervalle  $T_k$  dans  $[0, 1]$ .

Dans le cas de la décomposition de fonctions bivariées, un hypercube est associé à un couple  $\mathbf{d}_{k_r} = (d_{k_r,1}, d_{k_r,2})$ . L'hypercube  $S_{k_r}(\mathbf{d}_{k_r})$  est associé à un intervalle de valeur  $T_{k_r}(\mathbf{d}_{k_r})$  par la fonction  $\xi$ , comme l'illustre la figure 2.3(b).

Une fois les fonctions internes  $\psi$  et  $\xi$  calculées, il reste à évaluer les fonctions externes  $g_n$ . Cependant, ces fonctions ne peuvent pas être déterminées directement et ne peuvent seulement être calculées que par un processus itératif. Sprecher propose la construction de  $r$  fonctions  $g_n^r$  dont la somme converge vers la fonction externe  $g_n$ . L'algorithme construit itérativement en trois étapes une fonction externe  $g_n^r$ . La fonction  $f_r$  définit l'erreur d'approximation, qui tend vers 0 quand  $r$  croît. L'algorithme est initialisé avec  $f_0 = f$  et  $r = 1$ .

### Première étape : détermination de la précision et construction du pavage

Avant tout calcul, nous déterminons la précision  $k_r$  pour l'étape courante. Pour deux coordonnées  $x_i$  et  $x'_i$  de deux vecteurs différents, appartenant à la même dimension  $i$  et espacées d'une distance donnée, la distance des images des deux vecteurs  $\mathbf{x}$  et  $\mathbf{x}'$  produites par  $f$  doit être inférieure au  $N^e$  de l'oscillation de  $f$ , *i.e.* :

$$\text{si } |x_i - x'_i| \leq \frac{1}{\gamma^{k_r}}, |f_{r-1}(x_1, \dots, x_d) - f_{r-1}(x'_1, \dots, x'_d)| \leq \epsilon \|f_{r-1}\|.$$

Une fois  $k_r$  déterminé, le pavage  $d_{k_r,1}^n, \dots, d_{k_r,d}^n$  est calculé par :

$$\forall i \in \llbracket 1, d \rrbracket, d_{k_r,i}^n = d_{k_r,i} + n \sum_{r=2}^{k_r} \frac{1}{\gamma^r}$$

**Deuxième étape : évaluation des fonctions internes  $\psi$  et  $\xi$**

Pour  $n$  de 0 à  $m$ , déterminer  $\psi(d_{k_r}^n)$  et  $\xi(d_{k_r,1}^n, \dots, d_{k_r,d}^n)$  avec les équations 2.2 et 2.3.

**Troisième étape : détermination de l'erreur d'approximation**

$\forall n \in \llbracket 0, m \rrbracket$ , calculer :

$$g_n^r \circ \xi(x_1 + bn, \dots, x_d + bn) = \frac{1}{m+1} \sum_{d_{k_r,1}^n, \dots, d_{k_r,d}^n} f_{r-1}(d_{k_r,1}, \dots, d_{k_r,d}) \theta_{d_{k_r}^n}(\xi(x_1 + bn, \dots, x_d + bn))$$

où  $\theta$  est définie dans la définition 2. Ensuite, calculer :

$$f_r(x_1, \dots, x_d) = f(x_1, \dots, x_d) - \sum_{n=0}^m \sum_{j=1}^r g_n^j \circ \xi(x_1 + bn, \dots, x_d + bn)$$

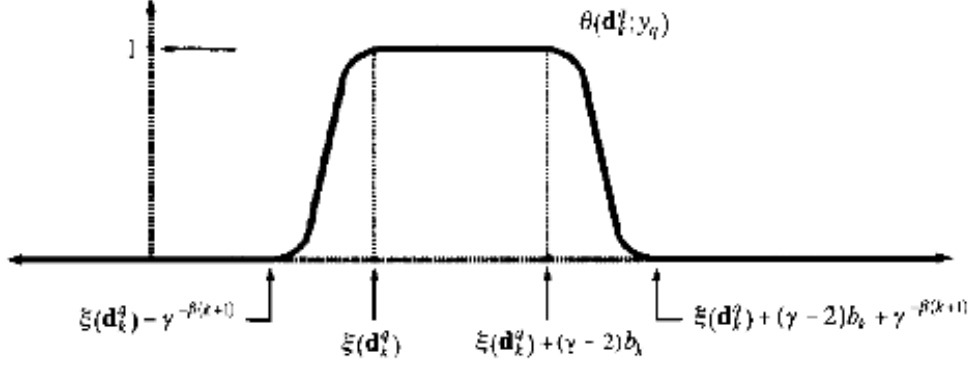
A la fin de la  $r^e$  itération, le résultat de l'approximation de  $f$  est donné par la somme des  $m+1$  couches de  $r$  fonctions  $g_n^r$  déterminées précédemment :

$$f \approx \sum_{n=0}^m \sum_{j=1}^r g_n^j \circ \xi(x_1 + bn, \dots, x_d + bn).$$

**Définition 2** Définition de la fonction  $\theta$  utilisée pour le calcul de la fonction  $g_n^r$ .

$$\theta_{d_k^n}(y_n) = \sigma\left(\gamma^{\frac{d^{k+1}-1}{d-1}}(y_n - \xi(d_k^n)) + 1\right) - \sigma\left(\gamma^{\frac{d^{k+1}-1}{d-1}}(y_n - \xi(d_k^n) - (\gamma - 2)c_k)\right), \quad (2.4)$$

avec  $y_n = \xi(x_1 + bn, \dots, x_d + bn)$



**Figure 2.4** – Représentation de la fonction  $\theta$ , d'après [Sprecher, 1997].

où  $\sigma$  est une fonction continue définie arbitrairement et vérifiant : 
$$\begin{cases} \sigma(x) \equiv 0, & \text{pour } x \leq 0 \\ \sigma(x) \equiv 1, & \text{pour } x \geq 1 \end{cases},$$
 et 
$$c_k = \sum_{r=k+1}^{\infty} \frac{1}{\gamma^{\frac{d^r-1}{d-1}}} \sum_{p=1}^n \lambda_p.$$

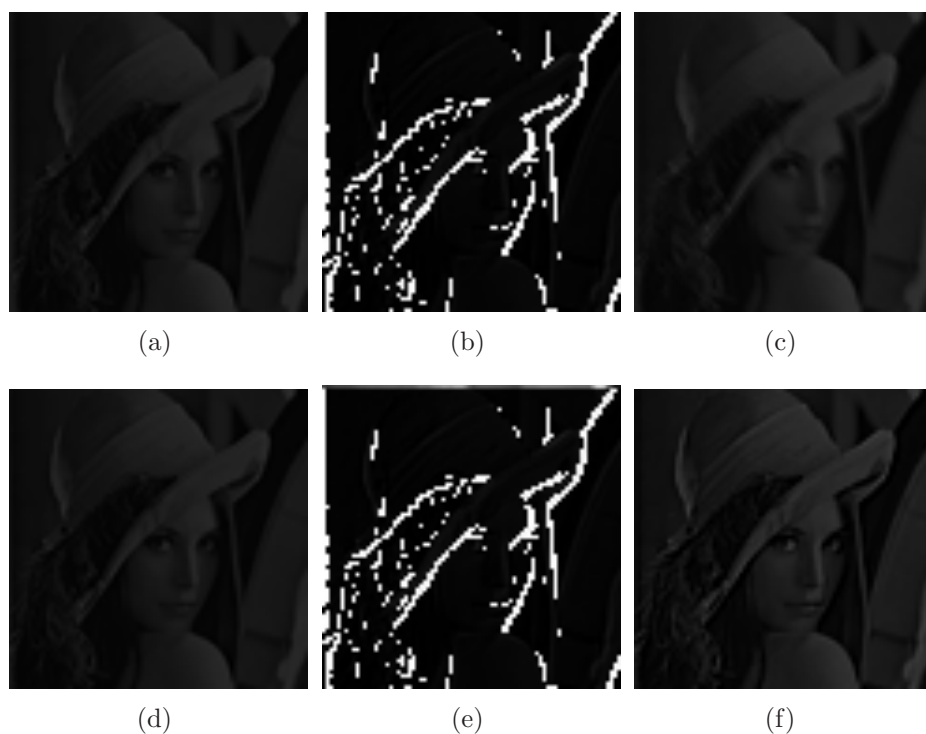
La Figure 2.4 illustre la fonction  $\theta$ . Cette fonction a une valeur constante sur un hypercube, et agit comme une fonction de mélange pour les fonctions externes lorsque les coordonnées considérées sont situés entre deux hypercubes.

**Remarque 1** *L'algorithme est très proche de l'énoncé du théorème, puisque la fonction interne est identique pour toute fonction  $f$  et que seules les fonctions externes sont adaptées à la fonction multidimensionnelle décomposée. Ces fonctions externes sont construites itérativement en prenant en compte les erreurs générées par les itérations antérieures. l'oscillation de la fonction d'erreur  $f_r$  tend vers 0 quand  $r$  croît, par conséquent, plus le nombre d'itérations est important, meilleure est l'approximation de la fonction  $f$  par les fonctions  $g_n$ .*

### 2.1.3 Application de la décomposition sur des images

Nous présentons ici les résultats de la décomposition appliquée à des images en niveaux de gris, pouvant être vues comme des fonctions bidimensionnelles  $f(x, y) = I(x, y)$ . La figure 2.5 représente deux couches issues de l'approximation obtenue après une et deux

itérations de l'algorithme de Sprecher. La somme de toutes ces couches donne l'approximation de l'image originale par l'algorithme. Les pixels blancs visibles sur les images 2.5(b) et 2.5(e) correspondent aux valeurs négatives de la fonction externe. La figure 2.6 montre deux reconstructions d'une même image de départ après une et deux itérations de l'algorithme. Les couches issues de la décomposition sont très similaires : chaque couche correspond à une fraction d'un sous-échantillon de la fonction  $f$ , légèrement traduite par un multiple de la valeur  $b$ . Dans le cas de fonctions bivariées, on observe que la reconstruction tend très rapidement vers l'image originale. On observe d'ailleurs très peu de différence entre la première et la seconde approximation sur les figures 2.6(b) et 2.6(c).



**Figure 2.5** – (a) et (b) Première couche ( $n = 0$ ) après une et deux itérations ( $r = 1, r = 2$ ) respectivement. (c) Somme de (a) et (b), reconstruction partielle pour la première couche. (d) et (e) La dernière couche ( $n = 5$ ) après une et deux itérations ( $r = 1, r = 2$ ) respectivement. (f) Somme de (d) et (e), reconstruction partielle pour la dernière couche.

### 2.1.4 Conclusion

L'algorithme proposé par Sprecher construit des fonctions internes et externes proches de la formulation du théorème : pour toute fonction  $f$  à décomposer, une fonction interne





**Figure 2.6** – (a) Image originale. (b) et (c) Reconstruction après une et deux itérations respectivement.

unique est utilisée, et seules les fonctions externes sont adaptées. Nous avons ensuite appliqué cet algorithme à la décomposition d’images en niveau de gris, considérées comme la représentation discrète d’une fonction bivariable. Nos résultats montrent que l’algorithme converge rapidement vers l’image originale. Cependant, le principal inconvénient de l’algorithme de Sprecher réside dans sa rigidité : les fonctions internes  $\psi$  ne peuvent pas être modifiées sans modifier la construction des fonctions externes. Or, dans la perspective d’étudier les possibles applications de la décomposition proposée par Kolmogorov, un algorithme flexible et facilement modifiable offrira davantage de possibilités d’adaptation. C’est pourquoi nous avons ensuite étudié l’algorithme d’Igel’nik et Parikh, qui ont proposé dans [Igel’nik and Parikh, 2003] un algorithme pour approcher les fonctions internes et externes, et qui permet en outre de disposer d’une représentation paramétrique continue de ces fonctions. Il permet notamment de faire varier le nombre de couches de pavage, de générer plusieurs fonctions internes, et de mettre à profit la continuité des fonctions pour différentes tâches d’interpolation et/ou de simplification. De plus, les approximations et définitions des fonctions monodimensionnelles sont indépendantes et modifiables en intervenant seulement localement dans l’algorithme.