Approximating Lipschitz continuous functions with GroupSort neural networks

Article de Ugo Tanielian, Maxime Sangnier, Gérard Biau

Présenté par : Ayoub Oulad Ali

Sorbonne Université

3 avril 2026

Contexte Mathématique et Notations

- **Espace ambiant :** L'espace ambiant $\mathbb{R}^d$ est supposé muni de la norme euclidienne $||\cdot||$. De plus, la notation $||\cdot||_\infty$ désigne la norme du supremum sur $\mathbb{R}^p$.

- **Fonctions 1-Lipschitziennes :** Pour un sous-ensemble $E \subseteq \mathbb{R}^d$, on note $Lip_1(E)$ l'ensemble des fonctions réelles 1-Lipschitziennes sur E.

$$Lip_1(E) = \{f : E \to \mathbb{R} : |f(x) - f(y)| \leq ||x - y||, (x, y) \in E^2\}$$

- **Normes matricielles :** Pour une matrice W de taille $p_1 \times p_2$:
  - Norme 2 : $||W||_2 = \sup_{||x||=1} ||Wx||$
  - Norme infinie : $||W||_\infty = \sup_{||x||_\infty=1} ||Wx||_\infty = \max_{i=1,\dots,p_1} \sum_{j=1}^{p_2} |w_{i,j}|$
  - Norme mixte $(2, \infty)$: $||W||_{2,\infty} = \sup_{||x||=1} ||Wx||_\infty$

- **Norme d'approximation :** L'espace des fonctions continues sur $[0, 1]^d$ est équipé de la norme uniforme.

$$||f - g||_\infty = \max_{x \in [0,1]^d} |f(x) - g(x)|$$

- **Classe de réseaux $\mathcal{D}_k$:** On note $\mathcal{D}_k = \{D_{k,\alpha} : \alpha \in \Lambda\}$ l'ensemble des fonctions de $\mathbb{R}^d$ dans $\mathbb{R}$ paramétrées par des réseaux de neurones de profondeur q utilisant l'activation GroupSort avec une taille de groupe $k \geq 2$.

L'Activation GroupSort et Propriétés de Représentation

L'Activation GroupSort (σ_k)

- **Principe :** Sépare l'entrée en groupes de k éléments et trie chaque groupe par ordre décroissant.
- **Exemple ($k = 2$) :**

$$\sigma_2(x_1, x_2) = (\max(x_1, x_2), \min(x_1, x_2))$$

- **Avantage :** C'est une fonction 1-Lipschitzienne qui préserve la norme du gradient de l'entrée.

Hypothèse 1 : Poids contraints

L'Hypothèse 1 impose des bornes strictes sur les matrices de poids : $||V_1||_{2,\infty} \leq 1$ et $\max(||V_2||_\infty, \ldots, ||V_q||_\infty) \leq 1$ et $\max(||c_q||_\infty, \ldots, ||c_1||_\infty) \leq K_2, K2 \geq 0$

Lemme 1 : Garantie Lipschitzienne

Sous l'Hypothèse 1 (bornes strictes sur les normes des poids $||V_i|| \leq 1$), pour tout $k \geq 2$, l'ensemble du réseau est 1-Lipschitz :

$$\mathcal{D}_k \subseteq Lip_1(\mathbb{R}^d)$$

Propriétés de $\mathcal{D}_k$: Compacité et Caractère Lipschitzien

Remarque : Compacité de $\mathcal{D}_k$

En combinant le Lemme 1 et le théorème d'Arzelà-Ascoli, on observe que sous l'Hypothèse 1, la classe $\mathcal{D}_k$ restreinte à un compact $K \subseteq \mathbb{R}^d$ est **compacte** dans l'ensemble des fonctions continues (pour la norme uniforme).

Éléments de preuve du Lemme 1 : $\mathcal{D}_k \subseteq Lip_1(\mathbb{R}^d)$

On écrit le réseau comme une composition $D_{k,\alpha}(x) = f_q \circ \cdots \circ f_1(x)$, avec $f_i(t) = \sigma_k(V_i t + c_i)$.

- **Première couche :** σ_k étant 1-Lipschitzienne, on a $||f_1(x) - f_1(y)||_\infty \leq ||V_1(x-y)||_\infty$. L'Hypothèse 1 impose $||V_1||_{2,\infty} \leq 1$, ce qui donne $||f_1(x) - f_1(y)||_\infty \leq ||x-y||$.
- **Couches cachées ($i \geq 2$) :** De la même manière, on obtient $||f_i(u) - f_i(v)||_\infty \leq ||V_i||_\infty ||u-v||_\infty$. Comme l'Hypothèse 1 exige que $\max(||V_i||_\infty) \leq 1$, l'inégalité devient $||f_i(u) - f_i(v)||_\infty \leq ||u-v||_\infty$.
- **Conclusion :** Par composition récursive de ces inégalités, on obtient finalement $|D_{k,\alpha}(x) - D_{k,\alpha}(y)| \leq ||x-y||$, prouvant que le réseau entier est bien 1-Lipschitzien.

Cadre d'Étude pour les Démonstrations

Simplification du cadre

Afin de rendre les démonstrations plus claires et d'illustrer les mécanismes clés sans alourdir les notations, nous nous placerons dans le cas particulier suivant pour les prochaines preuves :

- **Dimension de l'espace $d = 1$:** Nous étudierons des fonctions définies sur la droite réelle $\mathbb{R}$.
- **Taille de groupe $k = 2$:** L'activation GroupSort opérera uniquement sur des paires de composantes.
- **Nombre de morceaux $m = 2^n$ (avec $n \geq 1$) :** Le nombre de fonctions affines considérées sera une puissance de 2, ce qui simplifie les démonstrations par récurrence (structure d'arbre binaire).

Fonctions Linéaires par Morceaux et Partitions

Définition 1 : Fonction m_f-linéaire par morceaux

Une fonction continue $f : \mathbb{R}^d \to \mathbb{R}$ est m_f-linéaire par morceaux ($m_f \geq 2$) s'il existe :

- Une partition $\Omega = \{\Omega_1, \ldots, \Omega_{m_f}\}$ de $\mathbb{R}^d$ en polytopes.
- Une collection de fonctions affines $l_1, \ldots, l_{m_f}$ telles que $f(x) = l_i(x)$ sur chaque Ω_i.

La partition fine $\tilde{\Omega}$

- On définit une seconde partition $\tilde{\Omega} = \{\tilde{\Omega}_1, \ldots, \tilde{\Omega}_{M_f}\}$ (avec $M_f \geq 1$).
- Sur chaque sous-domaine $\tilde{\Omega}_k$, le **signe des différences** $l_i - l_j$ ne change pas.
- $\tilde{\Omega}$ est plus fine que Ω ($\tilde{\Omega}_i \subseteq \Omega_j$), ce qui implique que $M_f \geq m_f$.

Lemme 2 : Bornes sur le nombre de sous-domaines M_f

$$m_f \leq M_f \leq \min\left(2^{m_f^2/2}, \left(\frac{m_f}{\sqrt{2}}\right)^{2d}\right)$$

Démonstration 1 : Représentation du Max et Min (Proposition 1)

Proposition 1 (Cas $m = 2^n$ et $k = 2$)

Soient $f_1, \ldots, f_m$ représentées par des réseaux de profondeur q et de tailles s_i. Alors $f = \max(f_1, \ldots, f_m)$ (et $\min$) peuvent être représentées par un réseau GroupSort de profondeur $q + \log_2(m)$ et de taille $\leq \sum s_i + m - 1$.

Idée de la preuve par récurrence sur n :

- **Initialisation ($n = 1$, $m = 2$) :** Le calcul de $\max(f_1, f_2)$ nécessite simplement d'ajouter une couche GroupSort (1 neurone), passant la profondeur à $q + 1$ et la taille à $s_1 + s_2 + 1$.
- **Hérédité ($n > 1$) :**
  - On divise les m fonctions en deux moitiés de taille $m/2 = 2^{n-1}$.
  - Par hypothèse de récurrence, $g_1 = \max(f_1, \ldots, f_{m/2})$ et $g_2 = \max(f_{m/2+1}, \ldots, f_m)$ sont calculées avec une profondeur $q + n - 1$.
  - La somme de leurs tailles maximales donne $\sum s_i + m - 2$.
  - On ajoute une dernière couche (1 neurone) pour calculer $\max(g_1, g_2)$.
  - **Bilan :** La profondeur finale est $q + n = q + \log_2(m)$ et la taille augmente de 1, donnant bien $\sum s_i + m - 1$.

Représentation Exacte : Le Théorème 1 (Cas $d = 1$)

Le rôle central de la représentation Min-Max

Toute fonction m_f-linéaire par morceaux f s'écrit exactement en combinant un nombre fini d'opérations $\max$ et $\min$ sur ses fonctions affines l_i:

$$f = \max_{1 \leq k \leq M_f} \min_{i \in S_k} l_i$$

où chaque S_k est un sous-ensemble d'indices et M_f correspond au nombre de sous-domaines de notre partition fine.

Théorème 1 (Adapté pour $d = 1$)

Soit $f \in Lip_1(\mathbb{R})$ une fonction m_f-linéaire par morceaux. Il existe un réseau GroupSort (vérifiant l'Hypothèse 1) qui représente f exactement.

- **Profondeur :** $\lceil \log_2(M_f) \rceil + \lceil \log_2(m_f) \rceil + 1$
- **Taille maximale :** $3m_f M_f + M_f - 1$

Amélioration des Bornes : Sous-domaines Convexes

Corollaire 1 : Domaines Convexes

Soit $f \in Lip_1(\mathbb{R}^d)$ une fonction m_f-linéaire par morceaux avec des sous-domaines convexes $\Omega_1, \ldots, \Omega_{m_f}$. Alors, il existe un réseau de neurones GroupSort (vérifiant l'Hypothèse 1) qui représente exactement f.

De plus, ses caractéristiques sont considérablement réduites :

- **Profondeur :** $2\lceil \log_2(m_f) \rceil + 1$
- **Taille maximale :** $3m_f^2 + m_f - 1$

Expressivité et Compromis Profondeur/Largeur

Remarque sur l'expressivité : Lemme 3 ($d = 1, k = 2$)

Un réseau GroupSort de profondeur q et de largeurs $\nu_1, \ldots, \nu_{q-1}$ sur la droite réelle génère **au plus** un nombre de sous-domaines linéaires égal à :

$$2^{q-2} \times \left(\frac{\nu_1}{2} + 1\right) \times \nu_2 \times \cdots \times \nu_{q-1}$$

Corollaire 2 : Borne inférieure sur la taille

Pour représenter une fonction m_f-linéaire par morceaux avec une profondeur q, la taille totale s du réseau est minorée par : $s \geq \frac{1}{2}(q-1)m_f^{\frac{1}{q-1}}$.

Théorème 2 : Approximation de Fonctions Continues

Le Problème d'Approximation

Soit une fonction $f \in Lip_1([0,1])$ (pas nécessairement linéaire par morceaux) et un seuil de tolérance $\epsilon > 0$. On cherche un réseau D tel que :

$$||f - D||_\infty \leq \epsilon$$

Résultat de Complexité (pour $d = 1$ et $k = 2$)

Pour garantir une erreur maximale de ϵ, le réseau GroupSort nécessite :

- **Profondeur :** $\mathcal{O}(\log_2(1/\epsilon))$
- **Taille maximale :** $\mathcal{O}(1/\epsilon^2)$

Généralisation à k quelconques : Impact de la taille du groupe

Le principe d'un $k > 2$

Jusqu'à présent, avec $k = 2$, l'activation GroupSort calculait le $\max$ et le $\min$ de paires de neurones. Pour k quelconque, la fonction σ_k trie un groupe entier de k valeurs par ordre décroissant.

Impact sur l'approximation à ϵ près (Théorème 3)

Si l'on fait varier k dynamiquement en fonction de la précision voulue (ex: on choisit $k \propto 1/\epsilon$), les bornes s'effondrent :

- **En dimension d :** La profondeur devient $\mathcal{O}(d^2)$, ce qui est **indépendant** du taux d'erreur ϵ.
- **En dimension 1 ($d = 1$) :** Pour $k = \lceil 1/\epsilon \rceil$, un réseau de profondeur constante 3 et de taille $\mathcal{O}(1/\epsilon)$ suffit pour approximer la fonction !

Expériences Numériques et Implémentation

Le défi de l'entraînement : L'Orthonormalisation

Pour que les garanties théoriques (réseau 1-Lipschitz) soient valables en pratique, il faut maintenir l'Hypothèse 1 ($||V_i|| \leq 1$) durant l'entraînement.

- **Méthode :** Après chaque étape de descente de gradient, on applique une méthode d'orthonormalisation (comme l'algorithme de Björck) sur les matrices de poids.
- Cela garantit que les transformations restent non-expansives sans pour autant figer l'apprentissage.

Résultats de l'article

- **Régression ($d = 1$) :** L'augmentation de la taille du groupe k ($k = 2, 5, 10$) accélère drastiquement la convergence et la qualité de l'approximation.
- **Distance de Wasserstein :** Meilleure estimation de la distance duale par rapport aux réseaux ReLU contraints.

Notre démarche : Reproduction

- Nous allons recréer ces expériences dans les **mêmes conditions** pour vérifier empiriquement le compromis profondeur / expressivité.
- *Objectif :* Observer la stabilité de l'orthonormalisation et l'impact réel de la taille de groupe k face au bruit.

Synthèse des Résultats Expérimentaux

Table: Impact de la profondeur sur GroupSort ($\nu = 50$, fonction 32-PWL sur $[-8, 8]$)

Profondeur q	**Norme uniforme ($\downarrow$)**	**Constante de Lipschitz (≈ 1)**
$q = 2$	≈ 0.248	≈ 0.136
$q = 8$	≈ 0.138	≈ 0.541
$q = 14$	≈ 0.083	≈ 0.744
$q = 20$	$\approx \mathbf{0.028}$	$\approx \mathbf{0.946}$

Valeurs lues sur la Figure 3 de Tanielian et al. (2021). Le réseau de profondeur 20 atteint une taille réelle de $20 \times 60 = 1200$, bien en deçà de la borne théorème 1 : $3 \times 32^2 + 32 - 1 = 3104$.

Supériorité Théorique : GroupSort vs ReLU

Représentation de fonctions m-linéaires par morceaux (PWL)

Pour représenter une fonction composée de $m = k^n$ morceaux en dimension d (sous-domaines convexes) :

- **ReLU :** Taille $O(d^2 m^2)$ — He et al. (2018, Theorem 5.2).
- **GroupSort (k) :** Taille $\dfrac{m^2 - 1}{k - 1}$ — Corollaire 3, profondeur $2\lceil \log_k m \rceil + 1$.
- *Conclusion :* Pour k grand, la taille est réduite d'un facteur $k - 1$ par rapport à $k = 2$.

Approximation de fonctions 1-Lipschitz sur $[0, 1]$ (Dimension 1)

Pour atteindre une précision ϵ :

Méthode	Profondeur	Taille	Référence
ReLU Adaptatif	6	$O\left(\frac{1}{\epsilon \ln(1/\epsilon)}\right)$	Yarotsky (2017)
GroupSort ($k = \lceil 1/\epsilon \rceil$)	$\mathbf{3}$	$\mathbf{O(1/\epsilon)}$	Proposition 5

Limites de l'Approche : La Malédiction de la Dimension

Théorème 2

Pour approcher toute $f \in \mathrm{Lip}_1([0,1]^d)$ à précision ϵ, la taille requise est :

$$O\left(\left(\frac{2\sqrt{d}}{\epsilon}\right)^{d^2}\right)$$

Cette dépendance en d^2 à l'exposant est **doublement exponentielle** en dimension.

Validation Expérimentale : Estimation de la Distance W_1

Protocole : Dualité de Kantorovich-Rubinstein

Maximisation de $\mathbb{E}_{x\sim P}[f(x)] - \mathbb{E}_{y\sim Q}[f(y)]$ sous contrainte $\|f\|_L \leq 1$. 40 paires de mélanges gaussiens bivariés à 4 composantes (Figure 4 et Annexe B.2).

Architecture	Profondeur q	LRE (↓)	Largeur enveloppe (↓)
ReLU (libre)	2	4.33	0.43
ReLU + Björck	2	2.49	0.35
GroupSort ($k = 2$)	2	**1.54**	**0.17**
GroupSort ($k = 2$)	5	**1.01**	**0.13**

LRE = Least Relative Error. Source : Figures 4 et 13, Tanielian et al. (2021).

- **Observation :** L'erreur de ReLU libre est 3× plus grande que celle de GroupSort ; ReLU contraint reste significativement moins précis.
- **Interprétation :** La propriété *Gradient Norm Preserving* du GroupSort est cruciale : elle préserve le signal discriminatif dans la maximisation duale.

Validation des résultats

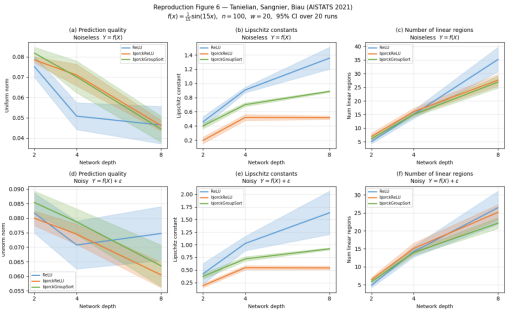


Figure 1 : résultats reconstruits

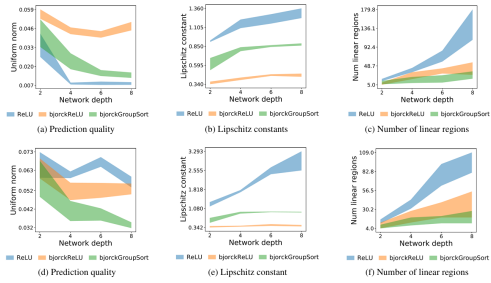


Figure 2 : Résultats originaux

Résultats supplémentaires

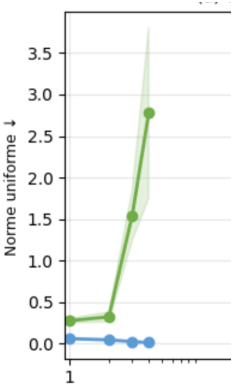


Figure 1 : Erreur norme infinie selon la dimension d = 1,..,5000

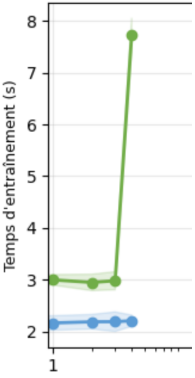


Figure 2 : Trade-off temps-dimension; d = 1,..,5000

Annexe A.1 — Lemme 1 : $\mathcal{D}_k \subseteq \mathrm{Lip}_1(\mathbb{R}^d)$

Énoncé

Sous l'Hypothèse 1, pour tout $k \geq 2$, tout réseau $D_{k,\alpha} \in \mathcal{D}_k$ est 1-Lipschitz.

Preuve par récurrence sur les couches. Soit $D_{k,\alpha}(x) = f_q \circ \cdots \circ f_1(x)$ avec $f_i(t) = \sigma_k(V_i t + c_i)$.

- **Couche 1 :** $\|f_1(x) - f_1(y)\|_\infty \leq \|V_1(x-y)\|_\infty \leq \|V_1\|_{2,\infty}\|x-y\|_2 \leq \|x-y\|_2$ (car σ_k est 1-Lip et $\|V_1\|_{2,\infty} \leq 1$ par Hyp. 1).
- **Couche $i \geq 2$:** $\|f_i(u) - f_i(v)\|_\infty \leq \|V_i\|_\infty \|f_{i-1}(u) - f_{i-1}(v)\|_\infty \leq \|f_{i-1}(u) - f_{i-1}(v)\|_\infty$ (car $\|V_i\|_\infty \leq 1$ par Hyp. 1).

Par composition : $|D_{k,\alpha}(x) - D_{k,\alpha}(y)| \leq \|x-y\|_2$. □

Annexe A.3 — Proposition 1 : Représentation du Max et Min

Énoncé

Soient $f_1, \ldots, f_m$ représentées par des réseaux GroupSort de profondeur q. Alors $\max(f_1, \ldots, f_m)$ est représentable en profondeur $q + \lceil \log_2 m \rceil$, taille $\leq s_1 + \cdots + s_m + m - 1$.

Preuve par récurrence sur n avec $m = 2^n$.

- **Base $n = 1$:** $\sigma_2(f_1, f_2) = (\max(f_1, f_2), \min(f_1, f_2))$ — une couche GroupSort suffit.
- **Hérédité :** Diviser en deux blocs ; $g_1 = \max(f_1, \ldots, f_{m/2})$ et $g_2 = \max(f_{m/2+1}, \ldots, f_m)$ sont réalisables en profondeur $q + n - 1$ par hypothèse. Concaténer une couche finale pour $\max(g_1, g_2)$.

Structure arborescente : la profondeur croît en $\log_2 m$, la taille n'augmente que de $m - 1$ neurones à comparer aux $8m - 4$ neurones nécessaires pour ReLU (Arora et al., 2018, Lemma D.3).

Annexe A.4 — Théorème 1 : Représentation Exacte des fonctions PWL

Énoncé

Toute $f \in \mathrm{Lip}_1(\mathbb{R}^d)$ qui est m_f-PWL est exactement représentable par un réseau GroupSort vérifiant l'Hypothèse 1, avec :

$$\text{profondeur} = \lceil \log_2 M_f \rceil + \lceil \log_2 m_f \rceil + 1, \qquad \text{taille} \leq 3 m_f M_f + M_f - 1.$$

Schéma de la preuve. He et al. (2018) donne $f = \max_{1 \leq k \leq M_f} \min_{i \in S_k} \ell_i$.

- Chaque ℓ_i : réseau affine à 1 neurone (Hypothèse 1 satisfaite).
- $\min_{i \in S_k} \ell_i$: Proposition 1 en $\lceil \log_2 m_f \rceil$ couches, taille $\leq 3m_f - 1$.
- $\max$ sur les M_f blocs : Proposition 1 en $\lceil \log_2 M_f \rceil$ couches supplémentaires.

Gain vs ReLU : la borne sur la taille est $O(m_f M_f)$ contre $O(d^2 m_f M_f)$ pour ReLU (He et al., 2018, Thm 5.2).

Annexe A.7 — Lemme 3 : Nombre de régions linéaires ($k = 2$)

Énoncé

Un réseau GroupSort($k = 2$) sur $\mathbb{R}$, de profondeur q et largeurs $\nu_1, \ldots, \nu_{q-1}$, paramètre une fonction PWL avec au plus

$$2^{q-2} \times \left(\tfrac{\nu_1}{2} + 1\right) \times \nu_2 \cdots \nu_{q-1}$$

sous-domaines linéaires.

Preuve par récurrence sur q.

- **Base $q = 2$:** chaque paire $\sigma_2(x_1, x_2)$ crée au plus 1 point d'inflexion $\Rightarrow \nu_1/2$ cassures, soit $\nu_1/2 + 1$ morceaux.
- **Hérédité :** l'entrée de chaque nœud de la couche i est une fonction à $\leq 2^{i-3}(\cdots)$ morceaux ; l'activation σ_2 double le compte. La couche finale somme ν_{q-1} fonctions.

Le nombre de régions croît **exponentiellement en q** et **polynomialement en ν** — analogue au Lemme 1.11 du cours (ReLU : $(2N)^{L-1}$ morceaux).

Annexe A.8 — Corollaire 2 : Borne inférieure taille/profondeur

Énoncé

Soit $f \in \mathrm{Lip}_1(\mathbb{R})$ une fonction m_f-PWL. Tout réseau GroupSort de profondeur q représentant f vérifie :

$$s \;\geq\; \tfrac{1}{2}(q-1)\, m_f^{1/(q-1)}.$$

Idée : le Lemme 3 donne $2^{q-2}(\nu_1/2+1)\cdots\nu_{q-1} \geq m_f$. Minimiser $s = \nu_1/2 + \cdots + \nu_{q-1}$ sous cette contrainte s'obtient en égalisant tous les termes — inégalité AM-GM.

Annexe A.9 — Théorème 2 : Approximation sur $[0,1]^d$

Schéma de la preuve (Cooper, 1995)

1. Tesseller $[0,1]^d$ en cubes de côté $\varepsilon/(2\sqrt{d}) \Rightarrow n = \lceil 2\sqrt{d}/\varepsilon \rceil^d$ cubes.
2. Un point par cube ; interpolation linéaire sur les simplexes de Delaunay. Nombre de sous-domaines : $m_f = O(n^{d/2})$, tous **convexes**.
3. Appliquer le Corollaire 1 : profondeur $O(d^2 \log_2(2\sqrt{d}/\varepsilon))$, taille $O((2\sqrt{d}/\varepsilon)^{d^2})$.

Annexe A.10 — Proposition 3 : Approximation sur $[0, 1]$

Construction

Pour $f \in \mathrm{Lip}_1([0,1])$, poser f_m l'interpolation linéaire aux $2^m + 1$ breakpoints $k/2^m$: $\|f - f_m\|_\infty \leq 2^{-m}$. Prendre $m = \lceil \log_2(1/\varepsilon) \rceil$, puis appliquer la Proposition 2.

Annexe A.11 — Proposition 4 : Extension à k quelconque

Idée (récurrence sur n, $m = k^n$)

Base $n = 1$ *:* $\max(f_1, \ldots, f_k)$ réalisable en profondeur $q + 1$, taille $s_1 + \cdots + s_k + 1$.
Hérédité : découper en k groupes de k^{n-1} fonctions, appliquer la récurrence sur chaque groupe, puis prendre le max des k résultats par une couche GroupSort(k) supplémentaire.
Taille finale : $s_1 + \cdots + s_m + \dfrac{m-1}{k-1}$.

Annexe A.12 — Corollaire 3

Énoncé

$f \in \mathrm{Lip}_1(\mathbb{R}^d)$, m_f-PWL à sous-domaines convexes, $m_f = k^n$. Profondeur : $2\lceil \log_k m_f \rceil + 1$. Taille : $\dfrac{m_f^2 - 1}{k - 1}$.

Preuve

He et al. (2018, Thm A.1) : $f = \max_{1 \le j \le m_f} \min_{i \in S_j} \ell_i$, $|S_j| \le m_f$.

1. Chaque $\min_{i \in S_j} \ell_i$: réseau de profondeur $\log_k(m_f) + 1$, taille $\frac{m_f - 1}{k-1}$ (variante Prop. 4).
2. $\max$ sur les m_f blocs : Prop. 4 à nouveau, profondeur $+ \log_k(m_f)$, taille $\times m_f$.

Annexe A.13 — Lemme 4 : Régions linéaires, k quelconque

Résultat

Un réseau GroupSort(k) sur $\mathbb{R}$, profondeur q, largeurs $\nu_1, \ldots, \nu_{q-1}$, paramètre une PWL avec au plus

$$k^{q-2} \times \left(\frac{(k-1)\nu_1}{2} + 1 \right) \times \nu_2 \cdots \nu_{q-1}$$

sous-domaines.

Preuve par récurrence. À profondeur 2, chaque groupe de k fonctions affines a au plus $\binom{k}{2}$ intersections $\Rightarrow \frac{(k-1)\nu_1}{2} + 1$ morceaux. Chaque couche supplémentaire multiplie ce compte par k.

Annexe A.14 — Théorème 3 : Résultat Dimension-Free

Preuve

Reprendre la preuve du Théorème 2 et remplacer le Corollaire 1 par le Corollaire 3 avec $k = \lceil 2\sqrt{d}/\varepsilon \rceil$:

$$\text{profondeur} = O(d^2), \qquad \text{taille} = O\left(\left(\frac{2\sqrt{d}}{\varepsilon}\right)^{d^2-1}\right).$$

La profondeur est **indépendante de** ε.

Annexe A.15 — Proposition 5 : Approximation sur $[0,1]$, k quelconque

Construction

Interpoler $f \in \mathrm{Lip}_1([0,1])$ aux $k^n + 1$ breakpoints i/k^n : $\|f - f_{k^n}\|_\infty \leq k^{-n}$. Prendre $n = \lceil \log_k(1/\varepsilon) \rceil$, appliquer le Corollaire 3.

Cas optimal $k = \lceil 1/\varepsilon \rceil$: $n = 1$, profondeur **3**, taille $O(1/\varepsilon)$.