Réseaux de neurones, approximation et application aux équations de transport linéaires

Jules Berry

31 décembre 2021

Table des matières

1 Introduction

Le point de départ de cette note est l'article [LP21] présentant une application de la théorie de l'approximation relative aux réseaux de neurones aux solutions des équations de transport paramétriques linéaires. Lors de la rédaction nous avons fait le choix de nous concentrer sur la présentation des résultats relatifs à l'approximation par réseaux de neurones en faisant passer les résultats relatifs aux équations de transport au second plan. En particulier on ne présente que le résultat principal de [LP21], c'est-à-dire le cas homogène. On pourra cependant y trouver le cas non-homogène ainsi que des équations de conservation. De plus nous n'aborderons pas ici les questions d'implémentation ou le processus d'apprentissage. On trouvera une introduction à ces questions dans chapitre 20 de [SSBD14].

Cette note est donc structurée de la manière suivante : la section 2 présente les définitions ainsi que quelques propriétés élémentaires relatifs à la manipulation de ceux-ci. On y présente aussi le cas unidimensionnel qui permet de se forger une intuition sur le comportements de ces objets. La section 3 contient les principaux résultats concernant les capacités d'approximation des réseaux de neurones. Il s'agit d'une part des théorèmes d'approximation universelle qui affirment que les réseaux de neurones sont capables d'approximer n'importe quelle fonction continue sur un compact. Ces théorèmes ne sont pas immédiatement liés aux résultats présentés dans [LP21] mais sont fondamentaux du point de vue de la théorie de l'approximation. D'autre part on y présente le théorème d'approximation obtenu par D. Yarotsky dans [Yar17] pour les fonctions régulières sur lequel repose [LP21]. Finalement on présente le résultat principal de [LP21] dans la section 4.

Nous avons aussi fait le choix de donner une présentation moins générale que ce qui est fait par les auteurs de [LP21]. En particulier nous nous sommes restreint à une classe de réseaux de neurones, dits *feedforward*, qui est un cas particulier de celle utilisé dans [LP21]. Ce choix permet d'alléger la présentation de la section 2 sans pour autant impacter la validité des résultats des sections 3 et 4. Dans le même esprit on énonce souvent sur $[0,1]^d$ ce qui est valable pour tout compact de $\mathbb{R}^d$. Enfin on trouvera un état de l'art récent de la théorie de l'approximation relative aux réseaux de neurones dans [DHP21].

2 Définitions

Dans cette section on présente les principales définitions et quelques résultats élémentaires relatifs aux réseaux de neurones.

2.1 Réseaux de neurones

On commence ici par définir les réseaux de neurones que nous considérerons ainsi que la réalisation qui leur est associée. Nous n'avons pas choisi la définition la plus générale, en particulier notre définition est un cas particulier de celle présentée dans [LP21], mais elle permet de simplifier la présentation sans impacter les différents résultats que nous présenterons ensuite.

Définition 2.1 (Réseau de neurones)**.** Soit $d, L \in \mathbb{N}$. Un réseau de neurones $\mathcal{N}$ de dimension d'entrée d et à L couches est une famille

$$\mathcal{N} = ((A_1, b_1), \dots (A_L, b_L))$$

où, pour $N_0 = d$, et $N_1, \dots N_L \in \mathbb{N}$, chaque A_j est une matrice réelle de taille $N_j \times N_{j-1}$ et $b_j \in \mathbb{R}^{N_j}$ pour $j = 0, \dots, L$. De plus on appelle $L = \mathrm{L}(\mathcal{N})$ la profondeur de $\mathcal{N}$ et $\mathrm{W}(\mathcal{N}) = \max(N_1, \dots N_{L-1})$ sa largeur. Enfin on appellera d les dimension d'entrée, N_L la dimension de sortie du réseau et $\sum_{i=0}^{L} N_i$ le nombre de neurones.

Définition 2.2 (Réalisation d'un réseau de neurones)**.** Soit $\sigma : \mathbb{R} \to \mathbb{R}$ la fonction partie entière $\sigma(x) = \max(0, x)$ [1] et soit $\mathcal{N}$ un réseau de neurones. On définit alors, avec les notations de la définition 2.1, la réalisation $\Phi = \mathrm{R}(\mathcal{N})$ de $\mathcal{N}$ comme étant la fonction $\Phi : \mathbb{R}^d \to \mathbb{R}^{N_L}$ donnée par la relation de récurrence

$$\begin{aligned} &x_0 = x \\ &x_j = \sigma(A_j x_{j-1} + b_j), \quad \text{pour } j = 1, \dots L-1, \\ &x_L = A_L x_{L-1} + b_L, \\ &\Phi(x) = x_L, \end{aligned}$$

où l'on applique la fonction σ composante par composante. Enfin en notant $N = (N_1, \dots, N_{L-1})$ on définit $\Upsilon_d^{N_L}(L, N)$ la classe des réalisations de réseaux de neurones $\Phi : \mathbb{R}^d \to \mathbb{R}^{N_L}$ où les $(N_j)_{1 \leq j \leq L-1}$ donnent les tailles des matrices et des vecteurs dans la définition 2.1.

Remarque 2.3.
- On appellera couche du réseau un couple (A_i, b_i) et N_i sa taille. De plus pour $i = 1, \dots L-1$ la couche sera dite cachée.
- Pour simplifier la présentation on considérera souvent des réseaux de neurones dont toutes les couches sont de même taille.

1. Dans la communauté des réseaux de neurones cette fonction est généralement appelée fonction ReLU pour *Rectified Linear Unit*. Il s'agit d'un cas particulier de ce que l'on appelle une fonction d'activation. Il est possible de raisonenr sur la base de fonctions d'activation plus générale mais il est devenu courant de se concentrer sur la fonction ReLU car il s'agit de la fonction la plus utilisée dans la pratique.

Pour essayer de comprendre le comportement d'un réseau de neurones il est intéressant de se ramener au cas où les dimensions d'entrée et de sortie sont 1. On commence par considérer les fonction de la forme $x \in \mathbb{R} \mapsto \sigma(ax + b)$ où a et b sont des réels. Il s'agit d'une fonction continue affine par morceaux a une seule rupture de pente. On se convainc alors facilement que $\sum_{j=1}^{N} \lambda_j \sigma(a_j x + b_j) - \gamma_j$ est une fonction continue affine par morceaux admettant au plus N ruptures de pente. De plus on peut aussi voir qu'en prenant N arbitrairement grand et en choisissant judicieusement les différents paramètres toute fonction continue affine par morceaux peut s'écrire sous cette forme. Plus précisément toute fonction continue affine par morceaux à M ruptures de pente peut être représentée par un réseau de neurones à une seule couche cachée le largeur M. On en déduit que , dans ce cas particulier, la classe des réseaux de neurones à une couche cachée contient l'ensemble des fonctions continues affines par morceaux. On sait que la qualité d'approximation de cette classe de fonction est essentiellement une fonction du nombre de ruptures de pente. Ce qui implique que la qualité de l'approximation fournie par le réseau à une couche cachée augmente avec le taille de la couche cachée.

Prolongeons alors le raisonnement aux fonction de la forme $x \in \mathbb{R} \mapsto \sigma(a_1\sigma(a_2 x + b_2) + b_1)$. C'est toujours une fonction continue affine par morceaux mais qui possède cette fois au plus deux ruptures de pente. En poursuivant ainsi on peut se convaincre que la réalisation d'un réseau de neurones est une fonction continue affine par morceaux dont le nombre de rupture de pente croît exponentiellement avec le nombre de couche. En revanche contrairement au cas à une seule couche cachée, un réseau de neurones dont la réalisation admet au plus M ruptures de pentes ne peut a priori pas représenter n'importe quelle fonction continue affine par morceaux à M ruptures de pente car les points de rupture sont liés par la relation de récurrence de la définition 2.2. Il a en revanche été démontré dans [DDF$^+$19] que pour tout M il existe une réseau de neurones de profondeur $L(M)$ qui peut représenter n'importe quel fonction continue affine par morceaux de $\mathbb{R}$ dans $\mathbb{R}$ admettant M ruptures de pente.

Enfin il est loin d'être évident que ces résultats en dimension un se généralisent en dimension plus grandes mais cela permet au moins d'avoir un début d'intuition sur le comportement des réseaux de neurones suivant les différents paramètres. A ce sujet on pourra consulter la section 3 de [DHP21].

2.2 Opérations sur les réseaux de neurones

On va ici définir deux opérations de bases sur les réseaux de neurones : la concaténation et la parallélisation. L'intérêt de ces opérations réside dans le fait que ces opérations s'interprèteront au niveau des représentations comme la composition et le produit direct. Cela nous permettra dans la suite de réduire l'approximation d'une fonction à l'approximation de briques plus simples.

Définition 2.4 (Concaténation)**.** Soient $L_1, L2 \in \mathbb{N}$,

$$\mathcal{N}_j = ((A_1^j, b_1^j), \dots, (A_{L_j}^j, b_j^j)), \quad j = 1, 2$$

deux réseaux de neurones tels que la dimension d'entrée de $\mathcal{N}_1$ soit égale à la dimension de sortie de $\mathcal{N}_2$ et $\Phi_j = \mathrm{R}(\mathcal{N}_j)$ leur réalisation respective. On définit alors la concaténation de $\mathcal{N}_1$ et $\mathcal{N}_2$, notée $\mathcal{N}_1 \odot \mathcal{N}_2$, comme étant le réseau de neurones de profondeur $L_1 + L_2 - 1$ donné par

$$\mathcal{N}_1 \odot \mathcal{N}_2 = ((A_1^2, b_1^2), \dots, (A_{L_2-1}^2, b_{L_2-1}^2), (A_1^1 A_{L_2}^2, A_1^1 b_{L_1}^2 + b_1^1), (A_2^1, b_2^1), \dots (A_{L_1}^1, b_{L_1}^1)).$$

L'intérêt de cette construction réside dans le proposition suivante.

Proposition 2.5. *Soit $\mathcal{N}_1, \mathcal{N}_2, \Phi_1, \Phi_2$ comme dans la définition 2.4. Alors on a*

$$\mathrm{R}(\mathcal{N}_1 \odot \mathcal{N}_2) = \Phi_1 \circ \Phi_2 \tag{1}$$

et de plus $\mathrm{W}(\mathcal{N}_1 \odot \mathcal{N}_2)) = \max(\mathrm{W}(\mathcal{N}_1), \mathrm{W}(\mathcal{N}_2))$ *et* $\mathrm{L}(\mathcal{N}_1 \odot \mathcal{N}_2) = \mathrm{L}(\mathcal{N}_1) + \mathrm{L}(\mathcal{N}_2) - 1$.

Démonstration. La relation (1) se vérifie immédiatement en exprimant la récurrence de la définition 2.2. □

Définition 2.6 (Parallélisation)**.** Soit $L \in \mathbb{N}$ et

$$\mathcal{N}_j = ((A_1^j, b_1^j), \dots, (A_L^j, b_j^j)), \quad j = 1, 2$$

deux réseaux de neurones de profondeur L et tous deux de dimension d'entrée $d \in \mathbb{N}$. On definit alors la parallélisation de $\mathcal{N}_1$ et $\mathcal{N}_2$, notée $\mathcal{N}_1 \otimes \mathcal{N}_2$, comme étant le réseau de neurones donné par

$$\mathcal{N}_1 \otimes \mathcal{N}_2 = ((\tilde{A}_1, \tilde{b}_1), \dots (\tilde{A}_L, \tilde{b}_L)),$$

où l'on a $\tilde{A}_1 = \begin{pmatrix} A_1^1 \\ A_1^2 \end{pmatrix}$, $\tilde{b}_1 = \begin{pmatrix} b_1^1 \\ b_1^2 \end{pmatrix}$, $\tilde{A}_j = \begin{pmatrix} A_j^1 & 0 \\ 0 & A_j^2 \end{pmatrix}$ et $\tilde{b}_j = \begin{pmatrix} b_j^1 \\ b_j^2 \end{pmatrix}$ pour $j = 2, \dots, L$. C'est alors un réseau de neurones de dimension d'entrée d de profondeur L.

De nouveau l'intérêt de la construction réside dans l'interprétation en terme de réalisation du réseau de neurones obtenu.

Proposition 2.7. *Soit $\mathcal{N}_1, \mathcal{N}_2$ deux réseaux de neurones comme dans la définition 2.6 et Φ_1, Φ_2 leur réalisation respective. On a alors*

$$\mathrm{R}(\mathcal{N}_1 \otimes \mathcal{N}_2) = (\Phi_1, \Phi_2) \tag{2}$$

et de plus $\mathrm{W}(\mathcal{N}_1 \otimes \mathcal{N}_2) \leq \mathrm{W}(\mathcal{N}_1) + \mathrm{W}(\mathcal{N}_2)$.

Démonstration. Ici aussi le résultat se déduit directement de la définition 2.2. □

En combinant parallélisation et concaténation on obtient facilement le résultat ci-dessous.

Proposition 2.8. *Soient $\mathcal{N}_1$ et $\mathcal{N}_2$ deux réseaux de neurones parallélisables, Φ_1, Φ_2 leur réalisation respective et λ_1, λ_2 deux réels. Il existe alors un réseau de neurones $\mathcal{N}$ tel que*

$$\mathrm{R}(\mathcal{N}) = \lambda_1 \Phi_1 + \lambda_2 \Phi_2.$$

Démonstration. Il suffit de voir qu'il existe une réseau de neurones de profondeur 0, de dimension d'entrée 2 et de dimension de sortie 1 tel que sa réalisation Ψ vérifie

$$\Psi(x) = \lambda_1 x_1 + \lambda_2 x_2.$$

On obtient alors le résultat par parallélisation et concaténation. □

Remarque 2.9. Par récurrence on peut facilement définir les concaténations et parallélisation de plus de deux réseaux. Dans ce cas les propositions ci-dessus se généralisent aisément.

3 Résultats d'approximation

Dans cette section on présente certains des résultats les plus importants concernant les propriétés d'approximation des réseaux de neurones. On commence par deux théorèmes dits d'approximation universelle qui affirment que les réseaux de neurones peuvent approximer n'importe quelle fonction raisonnable. Ces résultats ne donnant en revanche que très peu d'information sur la complexité du réseau nécessaire à l'approximation d'une fonction donné, on présente ensuite un résultat qui corrige cette lacune en exploitant une hypothèse de régularité sur la fonction à approximer.

3.1 Théorèmes d'approximation universelle

La première question qu'il est naturel de se poser lorsque l'on considère n'importe quelle méthode d'approximation est de déterminer la classe des fonctions qu'elle permet d'approximer. Plus particulièrement on peut se demander si, étant donnée une fonction continue, il existe un réseau de neurones dont la réalisation approxime cette fonction avec une précision arbitraire. Les théorèmes d'approximation universelle sont une réponse positive à cette question. Il existe plusieurs résultats de ce type mais nous n'en présenterons que deux ici qui sont en un sens le dual l'un de l'autre. Le premier que l'on présente, et le plus ancien, affirme que toute fonction continue peut être approximée par un réseau de neurones à une seule couche cachée, dans ce cas on permet

au réseau d'être arbitrairement large. Le second théorème fixe la largeur du réseau et affirme l'existence d'un réseau de neurones dont la profondeur peut cette fois être arbitrairement grande.

Théorème 3.1 (Premier théorème d'approximation universelle)**.** *Soit $f \in \mathcal{C}([0,1]^d, \mathbb{R}^n)$ et $\epsilon > 0$. Alors il existe un réseau de neurones à une seule couche cachée tel que sa réalisation Φ vérifie*

$$\|f - \Phi\|_{\mathcal{C}([0,1]^d, \mathbb{R}^n)} < \epsilon.$$

Démonstration. On peut se contenter de démontrer le résultat dans le cas où $f \in \mathcal{C}(\mathbb{R}^d, \mathbb{R})$, on obtient alors le résultat général par parallélisation. On considère

$$\Upsilon_1^1(1, \infty) = \left\{ \sum_{j=1}^{N} c_j \sigma(a_j x + b_j) : N \in \mathbb{N}^*,\ (a_j, b_j, c_j) \in \mathbb{R}^3,\ 1 \leq j \leq N \right\}$$

qui s'identifie avec la classe des réseaux de neurones dont l'entrée et la sortie sont de dimension un, ayant une seule couche cachée et une largeur arbitrairement grande. On sait que $\Upsilon_1^1(1, \infty)$ contient toutes les fonctions continues affines par morceaux de $\mathbb{R}$ dans $\mathbb{R}$. On sait que ces fonction sont denses dans les foctions continues, on en déduit donc que $\mathcal{C}(\mathbb{R}, \mathbb{R}) \subset \overline{\Upsilon_1^1(1, \infty)}$. On considère aussi l'ensemble de fonctions

$$\mathcal{R} = \text{Vect} \left\{ x \in [0,1]^d \mapsto g(\theta \cdot x) : \theta \in \mathbb{R}^d,\ g \in \mathcal{C}(\mathbb{R}, \mathbb{R}) \right\}.$$

Ce dernier contient en particulier les polynômes trigonométriques sur $[0,1]^d$, ce qui implique que $\mathcal{C}([0,1]^d, \mathbb{R}) \subset \overline{\mathcal{R}}$ d'après la théorie des séries de Fourier. On est alors en mesure de conclure de la manière suivante : pour $f \in \mathcal{C}([0,1]^d, \mathbb{R})$ et $\epsilon > 0$ il existe m fonctions $g_i \in \mathcal{C}(\mathbb{R}, \mathbb{R})$ et m paires $(d_i, \theta_i) \in \mathbb{R} \times \mathbb{R}^d$, $1 \leq i \leq m$ tels que

$$\|f(x) - \sum_{i=1}^{m} d_i g_i(\theta_i \cdot x)\|_{\mathcal{C}([0,1]^d, \mathbb{R})} < \frac{\epsilon}{2}.$$

De plus il existe pour chaque i, $1 \leq i \leq m$, des réels λ_j, a_j, b_j, $1 \leq j \leq p_j$ tels que

$$\|g_i - \sum_{j=1}^{p_i} \lambda_j \sigma(a_j \cdot - b_j)\|_\infty < \frac{\epsilon}{m}.$$

On a alors

$$\|f(x) - \sum_{i=1}^{m} \sum_{j=1}^{p_i} \lambda_j d_i \sigma(a_j \theta_i \cdot x - b_j)\|_{\mathcal{C}([0,1]^d, \mathbb{R})} < \epsilon.$$

En particulier, si l'on note $p = \sum_{i=1}^{m} p_j$, cela signifie qu'il existe $\Phi \in \Upsilon_d^1(1, p)$ telle que

$$\|f - \Phi\|_{\mathcal{C}([0,1]^d, \mathbb{R})} < \epsilon.$$

□

Remarque 3.2. Ce résultat peut être généralisé à des fonctions d'activations plus générales, voir [LLPS93] ou [Pin99] dont notre démonstration est largement inspirée.

On a alors le corollaire suivant.

Corollaire 3.3 ([LLPS93]). *Soit μ une mesure borélienne finie sur $[0,1]^d$, absolument continue par rapport à la mesure de Lebesgue. Soit $f \in L^p([0,1]^d, \mu)$, $1 \leq p < \infty$, et $\epsilon > 0$, alors il existe une réseau de neurones à une seule couche cachée dont la réalisation Φ vérifie*

$$\|f - \Phi\|_{L^p([0,1]^d,\mu)} < \epsilon.$$

On termine cette section en énonçant le théorème dual de 3.1.

Théorème 3.4 (Second théorème d'approximation universelle, [Han19]). *Soit $f \in \mathcal{C}([0,1]^d, \mathbb{R}_+)$ telle que $\|f\|_{\mathcal{C}([0,1]^d,\mathbb{R}_+)} = 1$ et $\epsilon > 0$. Alors il existe un réseau de neurones de largeur $d+2$ tel que sa réalisation Φ vérifie*

$$\|f - \Phi\|_{\mathcal{C}([0,1]^d,\mathbb{R}_+)} < \epsilon.$$

Remarque 3.5. Le résultat 3.4 démontré dans [Han19] n'est pas exactement l'analogue du théorème 3.1 mais on peut l'obtenir en considérant les bonnes parallélisations et combinaisons linéaires.

3.2 Approximation dans $W^{k,\infty}$

Les théorèmes 3.1 et 3.4 affirment qu'il existe un réseau de neurones approximant n'importe quelle fonction continue à une précision arbitraire. Ces résultats ne fournissent en revanche aucune borne sur la complexité du réseau en fonction de la précision. L'obtention d'une telle borne est l'objectif de cette section mais demande de faire une hypothèse supplémentaire sur la régularité de la fonction cible. On va considérer l'ensemble suivant :

$$F_d^k(\mathbb{R}^n) = \left\{ f \in W^{k,\infty}([0,1]^d, \mathbb{R}^n),\ \|f\|_{W^{k,\infty}([0,1]^d,\mathbb{R}^n)} \leq 1 \right\}.$$

Théorème 3.6 (Yarotsky, [Yar17]). *Soit $f \in F_d^k(\mathbb{R})$ et $\epsilon \in (0,1)$. Alors il existe une constante C, ne dépendant que de d et de k, et un réseau de neurones tel que son nombre de couches cachées soit borné supérieurement par $C(\ln(1/\epsilon)+1)$ et son nombre de neurones par $C\epsilon^{-d/k}(\ln(1/\epsilon)+1)$ et tel que sa réalisation Φ vérifie*

$$\|f - \Phi\|_{L^\infty([0,1]^d)} < \epsilon.$$

Démonstration. On ne reproduit pas ici la démonstration de [Yar17] mais on en présente les étapes principales. L'idée de la démonstration repose sur le lemme suivant.

Lemme 3.7. *Soit $M > 0$ et $\epsilon \in (0,1)$. Il existe alors un réseau de neurones $\mathcal{N}$ dont la réalisation $\tilde{\times} : \mathbb{R}^2 \to \mathbb{R}$ vérifie*

1. *pour tout $x, y \in [-M, M]$ on a $|\tilde{\times}(x,y) - xy| < \epsilon$,*
2. *si $x = 0$ ou $y = 0$, alors $\tilde{\times}(x,y) = 0$,*
3. *il existe deux constantes C_1 et $C_2 = C_2(M)$ telles que $\mathrm{L}(\mathcal{N}) \leq C_1 \ln(1/\epsilon) + C_2$ et de même pour son nombre de neurones.*

Ce qu'il faut retenir est que le lemme affirme qu'il est possible d'approximer le produit usuel par un réseau de neurones. On construit ensuite une partition de l'unité de $[0,1]^d$ de la manière suivante. Pour tout $N \in \mathbb{N}$ et $m \in 1, \ldots, N^d$ on pose

$$\phi_m(x) = \prod_{k=1}^{d} \psi\left(3N\left(x_k - \frac{m_k}{N}\right)\right)$$

où

$$\psi(x) = \begin{cases} 1, & |x| < 1, \\ 0, & 2 < |x|, \\ 2 - |x|, & 1 \leq |x| \leq 2. \end{cases}$$

Le support de ϕ_m est alors inclus dans $\{x : |x_k - m_k/N| < 1/N \quad \forall k\}$. On approche alors $f \in F_d^k$ par son polynôme de Taylor P_m à l'ordre $k-1$ au point m/N et on définit $f_1 = \sum_m \phi_m P_m$. On peut alors démontrer qu'il existe $N \in \mathbb{N}$ tel que

$$\|f - f_1\|_{L^\infty} < \frac{\epsilon}{2}.$$

On s'est donc ramené à approximer la fonction f_1. Or celle-ci est une combinaison linéaire du produit de ϕ_m avec un monôme de la forme $(x - m/N)^j$ avec $j \in 0, \ldots, n-1^d$. On remarque alors que les termes du produit dans la définition de ϕ_m sont représentables de manière exacte par un réseau de neurones sur l'intervalle $[0,1]$ et de même pour $t \mapsto t - m_k/N$. On obtient alors le résultat en quantifiant la complexité de ces réseaux et en appliquant autant de fois que nécessaire le lemme 3.7. □

Remarque 3.8.
— Comme on s'y attend le nombre de neurones nécessaire à l'approximation décroît lorsque la régularité de la fonction augmente à dimension constante. De même on constate que la situation idéale de produit lorsque la régularité k est très grande devant la dimension d.

— Si l'on considère une architecture dont toutes les couches cachées contiennent le même nombre de neurones, alors les couches du réseau permettant l'approximation dans le théorème 3.6 ont $\epsilon^{-d/k}$ neurones chacune.

— On trouvera une généralisation de ce résulats à d'autres espaces de Sobolev dans [GKP20].

Par parallélisation on obtient immédiatement le corollaire ci-dessous.

Corollaire 3.9. *Soit $f \in F_d^k(\mathbb{R}^n)$ et $\epsilon \in (0,1)$. Alors il existe une constante C, ne dépendant que de d et de k, et un réseau de neurones, dont toutes les couches contiennent le même nombre de neurones, tel que son nombre de couches cachées soit borné supérieurement par $C(\ln(1/\epsilon)+1)$, tel que son nombre de neurones par couche soit donné par $n\epsilon^{-d/k}$ et tel que sa réalisation Φ vérifie*

$$\|f-\Phi\|_{L^\infty([0,1]^d,\mathbb{R}^n)} < \epsilon.$$

On généralise les informations obtenues par l'intermédiaire du théorème 3.6 en introduisant la notion ci-dessous.

Définition 3.10. On dira qu'un fonction $f \in L^\infty([0,1]^d,\mathbb{R}^n)$ est r-approximable pour une réel $r > 0$ si pour tout $\epsilon \in (0,1)$ il existe une constate $C > 0$ et un réseau de neurones $\mathcal{N}$ vérifiant

1. le nombre de couches de $\mathcal{N}$ est borné supérieurement par $C(\ln(1/\epsilon)+1)$;
2. chaque couche contient a plus $\epsilon^{-1/r}$ neurones;
3. la réalisation Φ de $\mathcal{N}$ vérifie $\|f-\Phi\|_{L^\infty([0,1]^d,\mathbb{R}^n)} < \epsilon$.

D'après le théorème 3.6 et le corollaire 3.9 on sait qu'une fonction appartenant à $W^{k,\infty}([0,1]^d,\mathbb{R}^n)$ est k/d-approximable.

Théorème 3.11. *Soit $f : [0,1]^m \to \mathbb{R}^n$ une fonction r-approximable lipschitzienne et $g : [0,1]^d \to [0,1]^m$ une fonction s-approximable. Alors leur composition $f \circ g$ est θ-approximable où $\theta = \max(r,s)$.*

Démonstration. Soit $\epsilon > 0$ et L la constante de Lipschitz de f. On note $\delta_1 = \frac{\epsilon}{2L}$ et $\delta_2 = \frac{\epsilon}{2}$. Par hypothèse il existe deux réseaux de neurones $\mathcal{N}_f$ et $\mathcal{N}_g$, dont on note Φ_f et Φ_g les réalisations, qui vérifient

$$\|f-\Phi_f\|_{L^\infty} \le \delta_1 \quad \text{et} \quad \|g-\Phi_g\|_{L^\infty} < \delta_2$$

avec $W(\mathcal{N}_f) \le c\epsilon^{-1/r}$ et $W(\mathcal{N}_g) \le c'\epsilon^{-1/s}$. Alors par concaténation il existe un réseau de neurones $\mathcal{N} = \mathcal{N}_f \odot \mathcal{N}_g$ tel que $W(\mathcal{N}) \le \max(\epsilon^{-1/r}, \epsilon^{-1/s})$ dont la réalisation est $\Phi_f \circ \Phi_g$. De plus

$$\begin{aligned}\|f\circ g - \Phi_f\circ\Phi_g\|_{L^\infty} &\le \|f\circ g - f\circ\Phi_g\|_{L^\infty} + \|f\circ\Phi_g - \Phi_f\circ\Phi_g\|_{L^\infty}\\ &\le L\frac{\epsilon}{2L} + \frac{\epsilon}{2} \le \epsilon.\end{aligned}$$

□

Remarque 3.12. On a supposé que la fonction g prenait ses valeurs dans $[0,1]^m$ pour rester cohérent avec le reste de la présentation. On peut bien sûr lever cette hypothèse en généralisant les résultats précédents à des compacts quelconques.

4 Application aux équations de transport linéaires

On va à présent appliquer les propriétés d'approximation des réseaux de neurones à l'approximation des solutions des équations de transport de la forme

$$\begin{cases} \partial_t u(t,x,\lambda) + a(t,x,\lambda) \cdot \nabla_x u(t,x,\lambda) = 0, \\ u(0,x,\lambda) = u_0(x), \end{cases} \tag{3}$$

où $t \in [0,T]$, $x \in \mathbb{R}^d$ et $\lambda \in [0,1]^p$ est un paramètre de dimension p. On verra que l'apport des réseaux de neurones est particulièrement pertinent lorsque p est grand. On va supposer $a \in \mathcal{C}^k([0,T] \times \mathbb{R}^d \times [0,1]^p, \mathbb{R}^m)$, $|a(t,x,\lambda)| \leq C(1+|x|)$ pour tout $(t,x,\lambda) \in [0,T] \times \mathbb{R}^d \times [0,1]^p$ et $u_0 \in \mathcal{C}^s(\mathbb{R}^d, \mathbb{R})$.

On peut alors associer un flot $X : (s,t,x,\lambda) \in [0,T] \times [0,T] \times \mathbb{R}^d \times [0,1]^p \mapsto X(s,t,x,\lambda)$ au système différentiel

$$\begin{cases} \dot{y}(s) = a(s, y(s), \lambda), \\ y(t) = x, \end{cases}$$

tel que $X \in \mathcal{C}^k([0,T] \times [0,T] \times \mathbb{R}^d \times [0,1]^p)$. Sous ces conditions le problème 3 admet une unique solution donnée par

$$u(t,x,\lambda) = u_0(X(0,t,x,\lambda)).$$

On a alors immédiatement que la solution vérifie $u \in \mathcal{C}^r$ avec $r = \min(k,s)$. En particulier $u \in W^{r,\infty}([0,T] \times \mathbb{R}^d \times [0,1]^p)$ et en appliquant le théorème 3.6 on obtient le résultat suivant.

Proposition 4.1. *Pour tout $\epsilon \in (0,1)$ il existe un réseau de neurones $\mathcal{N}$, dont toutes les couches cachées ont le même nombre de neurones, et une constante $C > 0$ tels que*

1. *la longueur de $\mathcal{N}$ est bornée supérieurement par $C(\ln(1/\epsilon) + 1)$,*
2. *la largeur de $\mathcal{N}$ est bornée supérieurement par $\epsilon^{-(d+p+1)/r}$,*
3. *la réalisation Φ de $\mathcal{N}$ vérifie*

$$\|u - \Phi\|_{L^\infty} < \epsilon.$$

Il faut alors remarquer que le nombre de neurones nécessaires à l'approximation devient grand lorsque la dimension de l'espace des paramètres p l'est. Cependant le théorème 3.11 permet d'apporter une solution partielle à ce problème. En effet en considérant la composition $u = u_0 \circ X$ on obtient le résultat suivant.

Proposition 4.2. *Si $u_0 \in \mathcal{C}^s$ et $X \in \mathcal{C}^k$ alors la solution u de 3 est θ-approximable avec $\theta = \max(s/d, k/(d+p+1))$.*

L'intérêt de ce résultat est que si le flot X est assez régulier, les quantités s/d et $k/(d+p+1)$ peuvent toutes deux être proche de 1 bien que p soit grand. Ce qui peut grandement réduire le nombre de neurones nécessaire à l'approximation en comparaison de la proposition précédente.

Références

[DDF+19] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova. Nonlinear approximation and (deep) ReLU networks. *Constructive Approximation*, 2019.

[DHP21] Ronald DeVore, Boris Hanin, and Guergana Petrova. Neural network approximation. *Acta Numerica*, 2021.

[GKP20] I. Gühring, G. Kutyniok, and P. Petersen. Error bounds for approximations with deep ReLu neural networks in $W^{s,p}$ norms. *Analysis and Applications*, 2020.

[Han19] Boris Hanin. Universal function approximation by deep neural nets with bounded width and ReLU activations. *Mathematics*, 2019.

[LLPS93] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 1993.

[LP21] F. Laakmann and P. Petersen. Efficient approximation of solutions of parametric linear transport equations by ReLU DNNs. *Adv Comput Math*, 2021.

[Pin99] Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 1999.

[SSBD14] Shai Shalev-Schwartz and Shai Ben-David. *Understanding machine learning, from theory to algorithms*. Cambridge University Press, 2014.

[Yar17] Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 2017.