

# TER M1 : NN and RKHS

Matthieu Denis

27 août 2021

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Reproducing Kernel Hilbert space (RKHS) et leurs applications</b>	<b>5</b>
2.1	Cadre théorique . . . . .	5
2.2	Applications des RKHS en machine learning . . . . .	7
2.2.1	Representer theorem . . . . .	7
2.2.2	Exemple 1 : Kernel Ridge Regression . . . . .	7
2.2.3	Exemple 2 : Support Vector Machine (SVM) . . . . .	8
2.2.4	Le Kernel Trick . . . . .	9
<b>3</b>	<b>Introduction : Réseau de neurone simple</b>	<b>10</b>
3.1	Lois suivant la largeur des couches $m$ . . . . .	10
3.2	Gradients . . . . .	11
3.3	Descente de gradient . . . . .	11
3.3.1	Choix de $\eta$ . . . . .	12
3.3.2	Ordres de grandeur des écarts relatifs . . . . .	12
3.3.3	Choix de $\alpha$ . . . . .	13
3.4	Conclusion . . . . .	14
<b>4</b>	<b>Généralisation à un cas particulier</b>	<b>15</b>
4.1	Objectif . . . . .	15
4.2	Cas simple . . . . .	15
4.3	Cas général . . . . .	16
4.3.1	Dérivées premières . . . . .	16
4.3.2	Dérivées secondes . . . . .	16
4.3.3	Conclusion . . . . .	16
4.4	Application . . . . .	16
<b>5</b>	<b>Résultats récents</b>	<b>17</b>
5.1	Cadre . . . . .	17
5.2	Bornes à horizon fini . . . . .	17
5.3	Interprétation . . . . .	18
<b>6</b>	<b>Conclusion</b>	<b>18</b>
<b>7</b>	<b>Annexe : Preuves</b>	<b>19</b>
7.1	Théorème : Existence et unicité du noyau reproduisant . . . . .	19
7.2	Théorème : Moore-Aronszajn . . . . .	19
7.2.1	Representer theorem . . . . .	20
7.3	Lois suivant la largeur des couches $m$ . . . . .	21
7.4	Ordre de grandeur des écarts relatifs . . . . .	22
7.5	Choix de $\alpha$ . . . . .	22

7.6	Cas simple . . . . .	23
7.7	Dérivées premières . . . . .	24
7.8	Dérivées secondes . . . . .	25
7.9	Bornes à horizon fini : preuve du théorème . . . . .	26

# 1 Introduction

Ce rapport présente le travail encadré de recherche réalisé durant cet été 2021 sous la direction de M. Giraud au laboratoire de Mathématiques d'Orsay. Je tiens à le remercier de m'avoir proposé ce sujet, encadré, guidé et apporté son expertise pendant l'entièreté de ce stage. On traitera dans ce rapport les méthodes à noyau et des résultats théoriques sur les réseaux de neurones (NN).

En machine learning, les méthodes à noyau sont très utiles car elles permettent de transformer des algorithmes linéaires en algorithmes non-linéaires (le fameux kernel trick). On présentera premièrement le cadre théorique ainsi que des théorèmes amenant au kernel trick.

On étudiera ensuite le comportement asymptotique en la largeur des couches des réseaux de neurones, dans le cas où l'on initialise les paramètres et normalise d'une certaine manière ces NN. D'abord avec un cas simple : un NN linéaire. Ensuite à certains NN à deux couches. Puis finalement, en reprenant les résultats d'une publication de Chizat, un cas général qui englobe les NN ainsi que d'autres modèles lisses sous certaines hypothèses.

On essaiera dans tous les cas de voir apparaître un comportement linéaire des réseaux de neurones lorsque la largeur des couches est grande.

## 2 Reproducing Kernel Hilbert space (RKHS) et leurs applications

### 2.1 Cadre théorique

Nous allons dans cette section s'intéresser aux RKHS, des espaces de Hilbert réels qui satisfont certaines propriétés, et qui ont des applications intéressantes en machine learning. Par exemple, on montrera un théorème qui nous permet de simplifier un problème de minimisation de risque empirique de dimension infini à un problème en dimension fini. Ou encore des applications dans plusieurs algorithmes de ML, les transformant d'algorithmes linéaires à non-linéaires à très faible prix, et d'autres encore.

Posons d'abord quelques définitions et résultats concernant les RKHS, avant de passer aux applications.

Soit  $X$  un ensemble quelconque,  $H$  un espace de Hilbert de fonctions réelles sur  $X$ , muni de l'addition point par point ainsi que de la multiplication par scalaire point par point. On introduit aussi une forme linéaire qui à chaque fonction de  $H$  l'évalue en un certain point  $x \in X$ ,

$$L_x : f \mapsto f(x) \quad \forall f \in H$$

**Définition 1** (RKHS). On dit d'un espace de Hilbert qu'il est un RKHS si  $\forall x \in X$ ,  $L_x$  est continue sur  $H$ , ou encore si  $L_x$  est bornée sur  $H$ , i.e

$$\forall x \in X, \exists M_x > 0, \forall f \in H \text{ t.q } |L_x(f)| := |f(x)| \leq M_x \|f\|_H$$

Dans ce qui suit,  $H$  sera un RKHS.

**Propriété 1** (Convergence en norme dans un RKHS implique convergence point par point). Soient  $f_n, f \in H$ . Si  $f_n \xrightarrow{H} f$ , alors  $\forall x \in X$ ,  $f_n(x) \rightarrow f(x)$ .

*Démonstration.*

$$\forall x \in X, |f_n(x) - f(x)| = |L_x(f_n) - L_x(f)| = |L_x(f_n - f)| \leq M_x \|f_n - f\|_H$$

■

**Définition 2** (Noyau / Kernel). Une fonction  $k : X \times X \rightarrow \mathbb{R}$  est un noyau si

$$\exists \phi : X \rightarrow H \text{ t.q } k(x, y) = \langle \phi(x), \phi(y) \rangle_H \quad \forall x, y \in X$$

**Propriété 2.** Tout noyau  $k$  est symétrique défini positif.

*Démonstration.*

- Symétrie : découle de la symétrie du produit scalaire.

- Défini positif :

$$\forall x_1, \dots, x_n \in X, \forall c_1, \dots, c_n \in \mathbb{R}$$

$$\sum_{i,j=1}^n c_i c_j k(x_i, x_j) = \left\langle \sum_{i=1}^n c_i \phi(x_i), \sum_{j=1}^n c_j \phi(x_j) \right\rangle_H = \left\| \sum_{i=1}^n c_i \phi(x_i) \right\|_H^2 \geq 0$$

■

**Définition 3** (Noyau reproduisant / Reproducing kernel). Une fonction  $k : X \times X \rightarrow \mathbb{R}$  est un noyau reproduisant de  $H$  si  $\forall x \in X, f \in H$  :

- $k(\cdot, x) \in H$
- $f(x) = \langle f, k(\cdot, x) \rangle_H$

**Propriété 3.** Tout noyau reproduisant  $k$  est un noyau.

*Démonstration.* En prenant  $f = k(\cdot, y) \in H$  pour un certain  $y \in X$  dans la définition, on a en particulier  $\forall x, y \in X, k(x, y) = \langle k(\cdot, x), k(\cdot, y) \rangle_H$ . Ici  $\phi(u) = k(\cdot, u) \forall u \in X$ . ■

Le théorème de représentation de Riesz est utile pour montrer la propriété suivante, on l'énoncera seulement.

**Théorème 1** (Théorème de représentation de Riesz). Soient :

- $H$  un espace de Hilbert réel, muni de son produit scalaire  $\langle \cdot, \cdot \rangle_H$
- $L \in H'$  une forme linéaire continue sur  $H$ .

Alors

$$\exists! g \in H, \forall f \in H, L(f) = \langle f, g \rangle_H$$

**Propriété 4** (Existence et Unicité). Si  $H$  est un RKHS, alors il existe un unique noyau reproduisant de  $H$ .

**Théorème 2** (Lien des deux visions).  $H$  est un RKHS si et seulement si il existe un unique noyau reproduisant de  $H$ .

*Démonstration.* •  $(\Rightarrow)$  Donné par la propriété 4

- $(\Leftarrow) \forall x \in X, f \in H$

$$|L_x(f)| = |f(x)| = |\langle f, k(\cdot, x) \rangle_H| \leq \|f\|_H \|k(\cdot, x)\|_H = \underbrace{\sqrt{k(x, x)}}_{M_x} \|f\|_H$$

$M_x$  ne dépendant pas de  $f$ , on a bien l'inégalité  $\forall x \in X, \exists M_x > 0, \forall f \in H$ , i.e  $H$  est un RKHS. ■

**Théorème 3** (Théorème de Moore-Aronszajn). Soit  $K$  un kernel. Alors il existe un unique espace de Hilbert  $H$  de fonctions sur  $X$  pour lequel  $K$  est un noyau reproduisant.

On a montré avec le le théorème 2 que si l'on a un RKHS  $H$ , alors il existe un unique noyau reproduisant  $k$  pour  $H$ , celui-ci étant aussi un noyau par la propriété 3. Et on a ensuite montré par le théorème 3 que si l'on a un noyau  $k$ , alors c'est un noyau reproduisant d'un RKHS  $H$ . Entre autre, on a montré l'équivalence entre plusieurs représentations pour un RKHS.

Quelques kernels classiques comme exemples :

- $k(x, y) := \langle x, y \rangle$ , son RKHS est  $H = \{\langle \cdot, \beta \rangle : \|\langle \cdot, \beta \rangle\|_H^2 = \|\beta\|^2\}$
- $k(x, y) := (\alpha \langle x, y \rangle + 1)^d$ ,  $\alpha \in \mathbb{R}$ ,  $d \in \mathbb{N}$
- $k(x, y) := \exp(\|x - y\|^2 / (2\sigma^2))$ ,  $\sigma > 0$
- $k(x, y) := \exp(\|x - y\| / \sigma)$ ,  $\sigma > 0$
- $k(x, y) := \sin(a(x - y)) / \pi(x - y)$ , son RKHS correspond aux fonctions de  $L^2(\mathbb{R})$  dont la transformée de fourier est à support dans  $[-a, a]$ .

## 2.2 Applications des RKHS en machine learning

Une application très importante des RKHS est le theorem, un théorème qui permet de transformer des problèmes d'optimisation de dimension infinie en dimension finie, que l'on pourra alors résoudre avec les méthodes d'optimisation numérique. On verra son application avec le Kernel Ridge Regression et avec les SVM.

### 2.2.1 Representer theorem

**Théorème 4** (Representer theorem). Soit  $k$  un kernel sur  $X$  et soit  $H$  son RKHS associée. Posons  $x_1, \dots, x_n \in X$  notre training sample. Regardons le problème d'optimisation suivant :

$$\min_{f \in H} J(f) := E(f(x_1), \dots, f(x_n)) + P(\|f\|_H^2)$$

Où  $P$  est une fonction croissante.

Alors si ce problème d'optimisation a (au-moins) une solution, il y a (au-moins) une solution de la forme

$$f = \sum_{i=1}^n \alpha_i \cdot k(\cdot, x_i)$$

De plus, si  $P$  est strictement croissante, alors toute solution a cette forme.

### 2.2.2 Exemple 1 : Kernel Ridge Regression

Ici,  $J(f) := \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_H^2$ , prenons  $k$  un kernel sur  $X$ . Le representer theorem nous dit que la solution de ce problème (sous couvert d'existence) est nécessairement de la forme

$$f = \sum_{i=1}^n \alpha_i \cdot k(\cdot, x_i)$$

Rappelons que :

$$\|f\|_H^2 = \left\langle \sum_{i=1}^n a_i k(\cdot, x_i), \sum_{j=1}^n a_j k(\cdot, x_j) \right\rangle_H = \sum_{i=1}^n \sum_{j=1}^n a_i a_j K(x_i, x_j)$$

Le problème d'optimisation

$$\min_{f \in H} J(f) := \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_H^2$$

est alors équivalent à

$$\min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^n (y_i - \sum_{j=1}^n \alpha_j \cdot k(x_i, x_j))^2 + \lambda \sum_{i=1}^n \sum_{j=1}^n a_i a_j K(x_i, x_j)$$

ou encore avec  $(K)_{ij} = k(x_i, x_j)$  (qui est symétrique)

$$\min_{\alpha \in \mathbb{R}^n} F(\alpha) := \|y - K\alpha\|_2^2 + \lambda \alpha^T K \alpha$$

Reste à résoudre ce problème de la même manière qu'une régression ridge simple :

$$\nabla_{\alpha} F(\alpha) = -2K(y - K\alpha) + 2\lambda K\alpha$$

$$\nabla_{\alpha} F(\alpha) = 0 \Leftrightarrow \alpha = (K + \lambda Id_n)^{-1} y$$

Notre prédicteur sera ainsi

$$f = \sum_{i=1}^n ((K + \lambda Id_n)^{-1} y)_i \cdot k(\cdot, x_i)$$

Ce qui est mieux par rapport à la régression pénalisée, c'est que notre prédicteur est une fonction non-linéaire de  $x$ , nous permettant d'aller chercher dans la classe des fonctions de la RKHS associée à  $k$  (et donc potentiellement avoir de meilleures prédictions).

### 2.2.3 Exemple 2 : Support Vector Machine (SVM)

Dans le cadre d'une SVM,  $J(f) := \frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i f(x_i)) + \frac{\lambda}{2} \|f\|_H^2$ . Prenons un kernel  $k$  sur  $X$ . Encore une fois, le representor theorem nous dit que la seule solution (si elle existe) est sous la forme

$$f = \sum_{i=1}^n \alpha_i \cdot k(\cdot, x_i)$$

On cherche alors à résoudre le problème suivant :

$$\min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^n \max(0, 1 - y_i \sum_{j=1}^n \alpha_j \cdot k(x_i, x_j)) + \frac{\lambda}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j K(x_i, x_j)$$



On peut montrer que le dual de ce problème est :

$$\min_{\gamma \in \mathbb{R}^n} - \sum_{i=1}^n \gamma_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \gamma_i \gamma_j y_i y_j k(x_i, x_j) \text{ t.q } 0 \leq \gamma_i \leq \frac{1}{n\lambda} \forall i \in \{1, \dots, n\}$$

avec  $\alpha_i = y_i \gamma_i \forall i \in \{1, \dots, n\}$ . On trouve la la solution du dual par des algorithmes d'optimisation quadratique.

Le prédicteur est donc :

$$f = \sum_{i=1}^n y_i \gamma_i \cdot k(\cdot, x_i)$$

On a remplacé l'estimateur linéaire de la SVM par un estimateur non-linéaire de  $x$ , sans changer la complexité de l'algorithme.

#### 2.2.4 Le Kernel Trick

Plus généralement, à tout algorithme qui utilise des produits scalaires, on peut les remplacer par un kernel. Ainsi on peut transformer rendre non-linéaires les algorithmes, en manipulant des vecteurs de dimensions infinie sans que cela pose problème, comme leur produit scalaire est égal au kernel. C'est ce qu'on appelle le "kernel trick", et il a une très grande importance dans les applications pratiques. Par exemple, ce que l'on a fait plus haut avec peut aussi être fait avec le kernel trick dans la preuve de la construction de la solution linéaire en remplaçant  $\langle \cdot, \cdot \rangle$  par  $\langle \phi(\cdot), \phi(\cdot) \rangle = k(\cdot, \cdot)$ .

### 3 Introduction : Réseau de neurone simple

Commençons par étudier un NN très simple : une fonction  $\Phi : (\mathbb{R}^m \times \mathbb{R}^{m \times m} \times \mathbb{R}^m) \times \mathbb{R} \rightarrow \mathbb{R}$  combinaison d'applications linéaires, sans non linéarités intermédiaires :

$$\Phi((\beta, A, u), x) := \frac{1}{m^\alpha} \beta^T \left( \frac{1}{m^\gamma} A \right) u x$$

On initialise  $\theta^0 := (\beta^0, A^0, u^0)$  de manière standard :  $\forall i, j \in \{1, \dots, m\}$ ,  $u_i^0, A_{ij}^0, \beta_i^0 \sim_{iid} N(0, 1)$ . Nous montrerons quelques propriétés asymptotiques en la largeur des couches, et sur l'évolution des paramètres lors du premier pas de la descente de gradient. De plus, on discutera des répercussions du choix de  $\alpha$ .

#### 3.1 Loix suivant la largeur des couches $m$

Ci dessous on regarde les lois des paramètres et du réseau de neurone à l'initialisation, cela nous donne une vue d'ensemble et une manière de choisir  $\alpha$  et  $\gamma$  pour la normalisation.

- Loi de  $\|u^0\|_2^2$  pour  $m$  grand

$$\|u^0\|_2^2 \sim m \text{ pour } m \text{ grand.}$$

- Loi de  $(\frac{1}{m^\gamma} A^0) u^0 x$  sachant  $u^0$

$$\text{La loi de } (\frac{1}{m^\gamma} A^0) u^0 x \text{ sachant } u^0 \text{ est } N(0_m, (\frac{x}{m^\gamma})^2 \|u^0\|_2^2 Id_m).$$

- Loi de  $(\frac{1}{m^\gamma} A^0) u^0 x$  pour  $m$  grand

On aura besoin du lemme suivant :

$$X_n \sim N(\mu_n, \sigma_n) \text{ avec } \mu_n \rightarrow \mu \text{ et } \sigma_n \rightarrow \sigma, \text{ alors } (X_n) \text{ converge en loi vers } X_\infty \sim N(\mu, \sigma)$$

Regardons la variance de  $((\frac{1}{m^\gamma} A^0) u^0 x)_i$  sachant  $u^0$  pour  $m$  grand : on utilise  $\|u^0\|_2^2 \sim m$  et elle est donc environ égale à  $x^2 m^{1-2\gamma}$ . Si  $\gamma < 1/2$  cette variance diverge vers l'infini pour  $m \rightarrow \infty$ . Si  $\gamma > 1/2$ , elle tendra vers 0 pour  $m \rightarrow \infty$  et sa loi sera constante égale à 0. Seul le choix  $\gamma = 1/2$  permet de stabiliser la variance quand  $m \rightarrow \infty$ . Dans ce cas là, on applique le lemme : on a  $((\frac{1}{m^\gamma} A^0) u^0 x)_i$  sachant  $u^0$  qui converge vers une  $N(0, x^2)$ . Comme cette loi est indépendante de  $u^0$ , on a donc que  $((\frac{1}{m^\gamma} A^0) u^0 x)_i$  converge en loi vers  $N(0, x^2)$ .

- Loi de  $\|(\frac{1}{m^\gamma} A^0) u^0 x\|_2^2$  pour  $m$  grand

$$\|(\frac{1}{m^\gamma} A^0) u^0 x\|_2^2 \sim m x^2.$$

- Loi de  $\frac{1}{m^\alpha}(\beta^0)^T x_2$  sachant  $x_2$ , avec  $x_2 = \left(\frac{1}{m^\gamma} A^0\right) u^0 x$

On a  $\frac{1}{m^\alpha}(\beta^0)^T x_2 | x_2 \sim N(0, \frac{1}{m^{2\alpha}} \|x_2\|_2^2)$

- Loi de  $\frac{1}{m^\alpha}(\beta^0)^T x_2$  pour  $m$  grand

$$\frac{1}{m^\alpha}(\beta^0)^T x_2 \sim N(0, x^2 m^{1-2\alpha})$$

Le réseau de neurone reste centré à l'initialisation, on remarque de même que la variance reste constante que lorsque  $\alpha = 1/2$ . On impose ce choix là pour  $\alpha$ .

### 3.2 Gradients

$$\nabla_\beta \Phi = \frac{x}{m^{\alpha+1/2}} A u$$

$$\nabla_u \Phi = \frac{x}{m^{\alpha+1/2}} A^T \beta$$

$$\nabla_A \Phi = \frac{x}{m^{\alpha+1/2}} \beta u^T$$

Etudions les ordres de grandeur des normes correspondantes à l'initialisation pour  $m$  grand : On va simplement utiliser les approximations données par la loi des grands nombres :  $\|u^0\| \simeq \sqrt{m}$ , et comme vu plus haut,  $\|A^0 u^0\| \simeq \sqrt{m} \|u^0\| \simeq m$ . Ainsi  $\|\nabla_\beta \Phi(\theta^0, x)\| \sim m^{-\alpha-1/2} \cdot m = m^{1/2-\alpha}$ .

$$\|\nabla_\beta \Phi(\theta^0, x)\| \sim m^{1/2-\alpha}$$

Exactement de la même manière, on aboutit à :

$$\|\nabla_u \Phi(\theta^0, x)\| \sim m^{1/2-\alpha}$$

Pour  $A$ , on prend la norme de Frobenius : la LGN nous dit que  $\|\beta^0(u^0)^T\|_F^2 \simeq m^2$  (car il y a  $m^2$  lois du  $\chi^2(1)$  dans  $\beta^0(u^0)^T$ ). Ainsi  $\|\beta^0(u^0)^T\|_F \simeq m$  et donc :

$$\|\nabla_A \Phi(\theta^0, x)\|_F \sim m^{1/2-\alpha}$$

### 3.3 Descente de gradient

On va étudier ici le premier pas de descente de gradient.

Posons une fonction de perte  $F : \mathbb{R} \rightarrow \mathbb{R}$  t.q  $F'(0) \neq 0$  et  $\Delta F := F(\Phi(\theta^1, x)) - F(\Phi(\theta^0, x))$ , avec  $\theta^1 := \theta^0 - \eta \nabla_\theta F(\Phi(\theta^0, x))$

Il semble honnête de prendre  $\eta$  dépendant de  $m$ , le produit scalaire final ayant plus de chance d'exploser en grande dimension. Prenons  $\eta := m^a$ ,  $a \in \mathbb{R}$

### 3.3.1 Choix de $\eta$

On veut que  $\Delta F$  ne diverge pas ni ne tende vers 0 lorsque  $m$  tend vers l'infini. Pour cela, on utilise l'approximation  $\Delta F \simeq \langle \Delta\theta, \nabla_\theta F(\Phi(\theta^0, x)) \rangle$ .

On a

$$\Delta F \simeq \langle -\eta \nabla_\theta F(\Phi(\theta^0, x)), \nabla_\theta F(\Phi(\theta^0, x)) \rangle = -\eta \|\nabla_\theta F(\Phi(\theta^0, x))\|^2$$

$$\nabla_\theta F(\Phi(\theta^0, x)) = \underbrace{F'(\Phi(\theta^0, x))}_{\text{constante en } m} \cdot \nabla_\theta \Phi(\theta^0, x)$$

$$\text{Or } \|\nabla_\theta \Phi(\theta^0, x)\|^2 = \|\nabla_\beta \Phi(\theta^0, x)\|^2 + \|\nabla_u \Phi(\theta^0, x)\|^2 + \|\nabla_A \Phi(\theta^0, x)\|^2$$

Donc pour  $m$  grand :

$$\begin{aligned} \Delta F &\simeq -\eta \|\nabla_\theta F(\Phi(\theta^0, x))\|^2 \\ &\sim \eta \|\nabla_\theta \Phi(\theta^0, x)\|^2 \\ &\simeq \eta (3 \cdot (m^{1/2-\alpha})^2) \\ &\simeq m^a \cdot m^{1-2\alpha} \end{aligned}$$

Ce qui impose le choix  $a = 2\alpha - 1$  pour que  $\Delta F$  reste constant lorsque  $m$  tend vers l'infini.

### 3.3.2 Ordres de grandeur des écarts relatifs

Pour cela introduisons  $\Delta\theta := \theta^1 - \theta^0$ . Remarquons que  $\Delta\theta = -\eta \nabla_\theta F(\Phi(\theta^0, x))$ .

$$\|\Delta u\| \sim \eta \|\nabla_u \Phi(\theta^0, x)\| \tag{1}$$

$$\sim m^{2\alpha-1} \cdot m^{1/2-\alpha} \tag{2}$$

$$\sim m^{\alpha-1/2} \tag{3}$$

Ce qui nous donne un ordre de grandeur de l'écart relatif :

$$\frac{\|\Delta u\|}{\|u^0\|} \sim m^{\alpha-1}$$

On a le même résultat pour l'écart relatif de  $\beta^0$  :

$$\frac{\|\Delta \beta\|}{\|\beta^0\|} \sim m^{\alpha-1}$$

Pour  $A$ , la LGN nous donne  $\|A^0\|_F \simeq m$  pour  $m$  grand, on a alors par les mêmes calculs :

$$\frac{\|\Delta A\|_F}{\|A^0\|_F} \sim m^{\alpha-3/2}$$

Concernant l'écart entrywise de  $A$ , on a  $|\Delta A_{ij}| \sim \eta |(\nabla_A \Phi(\theta^0, x))_{ij}| \sim m^{2\alpha-1} \cdot m^{-1/2-\alpha} \cdot 1 \sim m^{\alpha-3/2}$  car  $|\beta^0(u^0)^T| \sim 1$ .  $|A_{ij}| \sim 1$ , donc :

$$\frac{|\Delta A_{ij}|}{|A_{ij}|} \sim m^{\alpha-3/2}$$

Maintenant avec la norme opérateur (voir annexe) :

$$\frac{|\Delta A|_{op}}{|A^0|_{op}} \sim m^{\alpha-1}$$

### 3.3.3 Choix de $\alpha$

On va voir dans cette section que le choix de  $\alpha$  est déterminant dans le comportement du réseau de neurone.

- $\alpha < 1$

Dans ce cas là, tous les écarts relatifs d'ordre  $m^{\alpha-1} \xrightarrow{m \rightarrow \infty} 0$ . On a donc pour  $m$  grand :

$$\|\Delta u\| \ll \|u^0\|$$

$$\|\Delta \beta\| \ll \|\beta^0\|$$

$$|\Delta A|_{op} \ll |A^0|_{op}$$

Le premier pas de descente de gradient n'a pas significativement fait bouger nos paramètres de l'initialisation.

Regardons maintenant les  $\Delta$  des gradients pour la première itération :

On peut montrer que (voir annexe)  $\|\Delta \nabla_u \Phi\| = \mathcal{O}(m^{-1/2})$  pour  $m$  grand.

De même,  $\|\Delta \nabla_\beta \Phi\| = \mathcal{O}(m^{-1/2})$  et  $\|\Delta \nabla_A \Phi\|_F = \mathcal{O}(m^{-1/2})$  pour  $m$  grand.

Ainsi :

$$\|\Delta \nabla_{\beta/u/A} \Phi\| = \mathcal{O}(m^{-1/2}) \stackrel{\alpha < 1}{\ll} m^{1/2-\alpha} \sim \|\nabla_{\beta/u/A} \Phi\|$$

Cela traduit un comportement linéaire lorsque  $\alpha < 1$  pour  $m$  grand.

On peut aussi remarquer que

$$\Delta \nabla_\theta F(\Phi) = F'(\Phi) \cdot \Delta \nabla_\theta \Phi$$

On a aussi :

$$\|\Delta \nabla_{\beta/u/A} F(\Phi)\| = \mathcal{O}(m^{-1/2}) \stackrel{\alpha < 1}{\ll} m^{1/2-\alpha} \sim \|\nabla_{\beta/u/A} F(\Phi)\|$$

Ainsi, lorsqu'on fait un pas de gradient, la variation du gradient est quasiment nulle. On utilise alors l'approximation  $\nabla_{\beta/u/A} F(\Phi(\theta^1, x)) \simeq \nabla_{\beta/u/A} F(\Phi(\theta^0, x))$ . On peut alors réutiliser les calculs pour l'étape suivante et ainsi de suite. On finit donc par avoir l'approximation  $\nabla_{\beta/u/A} F(\Phi(\theta^t, x)) \simeq \nabla_{\beta/u/A} F(\Phi(\theta^0, x))$  après  $t$  pas de gradient.

On peut aussi montrer que

$$F(\Phi(\theta^T, x)) = F(\Phi(\theta^0, x)) + \langle \theta^T - \theta^0, \nabla_{\theta} F(\Phi(\theta^0, x)) \rangle + \mathcal{O}(m^{-1/2})$$

On apprend donc un modèle linéaire relatif aux features  $\nabla_{\theta} F(\Phi(\theta^0, x))$ , c'est-à-dire qu'après la transformation  $x \rightarrow \nabla_{\theta} F(\Phi(\theta^0, x))$ , on est linéaire. On fait donc face à un RKHS de noyau (par définition)

$$k(x, y) = \langle \nabla_{\theta} F(\Phi(\theta^0, x)), \nabla_{\theta} F(\Phi(\theta^0, y)) \rangle \xrightarrow{LGN} \mathbb{E}[\langle \nabla_{\theta} F(\Phi(\theta^0, x)), \nabla_{\theta} F(\Phi(\theta^0, y)) \rangle] \quad \forall x, y \in \mathbb{R}$$

Le noyau dépend seulement de l'architecture du NN et de l'initialisation, il n'y a donc pas de feature learning.

- $\alpha = 1$

Dans ce cas là, comme vu plus haut, les variations relatives des paramètres ne sont plus négligeable pour  $m$  grand, et on observe plus de RKHS, il y a bien feature learning.

### 3.4 Conclusion

Ici on a fait le calcul dans le cas le plus simple possible pour voir apparaître le phénomène de linéarité asymptotique. Ce n'est pas contre intuitif pour le moment car le réseau de neurone est linéaire. Mais ci-dessous, on retrouve les mêmes résultats pour un réseau de neurone non linéaire (i.e un réseau de neurone utile), on ne s'y attend pas forcément.

## 4 Généralisation à un cas particulier

### 4.1 Objectif

Dans cette partie, on va étudier ce comportement linéaire autour de l'initialisation pour un cas particulier d'un réseau de neurone à 2 couches :

$$\frac{1}{\sqrt{m}} \sum_{j=1}^m a_j \cdot \sigma(\langle \mathbf{b}_j, x \rangle)$$

Où  $a_j \in \mathbb{R}$ ,  $x \in \mathbb{R}^d$ , et  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  une fonction non-linéaire. C'est une moyenne coefficientée renormalisée de la couche cachée. Les  $a_j$  et  $\mathbf{b}_j$  seront initialisés gaussiennement. Pour faciliter les calculs, on considèrera  $u_j := (a_j, \mathbf{b}_j)$  les paramètres de ce NN.

### 4.2 Cas simple

Soit  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  et

$$g(u) := \frac{1}{m} \sum_{j=1}^m \phi(u_j)$$

On suppose l'initialisation gaussienne :  $u_j^0 \sim_{iid} N(0, 1)$ .

Etudions le comportement de  $f(u) := \sqrt{m}g(u)$  autour de  $u_j^0$  pour  $m$  grand :

On peut montrer (voir annexe) que

$$\|\nabla g(u)\|^2 \simeq \frac{1}{m} \mathbb{E}((\phi'(u_j))^2)$$

et que

$$\|D^2 g(u)\|_{op} \leq \frac{|\phi''|_\infty}{m}$$

On peut à présent regarder le comportement pour  $m$  grand de  $f(u) := \sqrt{m}g(u)$ . Pour tout  $h$ , on a :

$$\begin{aligned} f(u^0 + h) &= f(u^0) + \sqrt{m} \left( \langle \nabla g(u), h \rangle + \mathcal{O}\left(\frac{\|h\|^2}{m}\right) \right) \\ &= f(u^0) + \sqrt{m} \|\nabla g(u)\| \left\langle \frac{\nabla g(u)}{\|\nabla g(u)\|}, h \right\rangle + \mathcal{O}\left(\frac{\|h\|^2}{\sqrt{m}}\right) \\ &\stackrel{m \rightarrow \infty}{\simeq} f(u^0) + \sqrt{\mathbb{E}((\phi'(u_j))^2)} \langle V_u, h \rangle + \mathcal{O}\left(\frac{\|h\|^2}{\sqrt{m}}\right) \text{ avec } V_u := \frac{\nabla g(u)}{\|\nabla g(u)\|} \text{ de norme 1.} \end{aligned}$$

On voit donc que notre réseau de neurone se comporte encore une fois linéairement pour  $m$  grand.

### 4.3 Cas général

Maintenant, considérons  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ . On va montrer le même résultat pour cette fonction  $\phi$  multivariée.

#### 4.3.1 Dérivées premières

$$\|Dg(u)\|_{op} = \|Dg(u)Dg(u)^T\|_{op}^{1/2} \simeq \frac{1}{\sqrt{m}} \|\mathbb{E}(D\phi(u_j^0)D\phi(u_j^0)^T)\|_{op}^{1/2}$$

#### 4.3.2 Dérivées secondes

$$\|D^2g(u)\|_{op} \leq \frac{1}{m} \max_{j=1 \dots m} \|D^2\phi(u_j)\|_{op}$$

#### 4.3.3 Conclusion

On peut donc regarder comme précédemment le comportement pour  $m$  grand de  $f(u) := \sqrt{m}g(u)$ . Pour tout  $h$ , on a :

$$f(u^0+h) \stackrel{m \rightarrow \infty}{\simeq} f(u^0) + \|\mathbb{E}(D\phi(u_j^0)D\phi(u_j^0)^T)\|_{op}^{1/2} \langle V_u, h \rangle + \mathcal{O}\left(\frac{\|h\|^2}{\sqrt{m}}\right) \text{ avec } V_u := \frac{Dg(u)}{\|Dg(u)\|_{op}} \text{ de norme 1}$$

On voit donc que notre réseau de neurone se comporte encore une fois linéairement pour  $m$  grand.

### 4.4 Application

Revenons à notre objectif initial : le comportement asymptotique autour de l'initialisation des paramètres de ce NN :

$$\frac{1}{\sqrt{m}} \sum_{j=1}^m a_j \cdot \sigma(\langle \mathbf{b}_j, x \rangle)$$

Ici, on a juste à prendre  $\phi(u_j) = a_j \cdot \sigma(\langle \mathbf{b}_j, x \rangle)$ .

Le réseau de neurone se comporte donc linéairement lorsque  $m$  tend vers l'infini. On a donc montré que le réseau de neurone non-linéaire se comporte asymptotiquement en le nombre de couches linéairement, un résultat surprenant : il n'y a donc pas de feature learning (on ne dépend que de l'initialisation) et on fait encore face à une RKHS.



## 5 Résultats récents

On regarde dans cette partie les résultats d'une publication de Lénaïc Chizat, Edouard Oyallon, et Francis Bach : "On Lazy Training in Differentiable Programming" [2]. Cette publication donne des résultats sur le comportement après un nombre fini de pas de gradient de modèles lisses ainsi que de leurs paramètres. On voit alors que lorsqu'on normalise d'une certaine manière, le modèle lisse se comporte comme son modèle linéarisé (que l'on définit ci-dessous) et que les paramètres restent proches de l'initialisation.

### 5.1 Cadre

Considérons un espace de paramètres  $\mathbb{R}^p$ , un espace de Hilbert  $\mathcal{F}$ , un modèle lisse  $h : \mathbb{R}^p \rightarrow \mathcal{F}$  (un NN par exemple) et une fonction de perte lisse  $R : \mathcal{F} \rightarrow \mathbb{R}_+$ . On veut minimiser la fonction objectif normalisée  $F_\alpha : \mathbb{R}^p \rightarrow \mathbb{R}_+$  :

$$F_\alpha(w) := \frac{1}{\alpha^2} R(\alpha h(w))$$

pour un certain  $\alpha \in \mathbb{R}_+$ .

On définit ensuite le modèle linéarisé autour de l'initialisation  $w_0$  :  $\bar{h}(w) := h(w_0) + Dh(w_0)(w - w_0)$  et sa fonction objectif normalisée  $\bar{F}_\alpha : \mathbb{R}^p \rightarrow \mathbb{R}_+$  :

$$\bar{F}_\alpha(w) := \frac{1}{\alpha^2} R(\alpha \bar{h}(w))$$

Pour prouver les résultats qui suivent, on aura besoin d'hypothèses supplémentaires :

**Hypothèses :**  $h$  est différentiable et de différentiel  $Dh$  localement Lipschitz (par rapport à la norme opérateur).  $R$  est différentiable et de gradient Lipschitz (par rapport à la norme de  $\mathcal{F}$ ).

**Flot de gradient de  $F_\alpha$  :** On étudie ci-dessous le flot de gradient de  $F_\alpha$ , qui est un chemin à temps continu  $(w_\alpha(t))_{t \geq 0}$  de paramètres dans  $\mathbb{R}^p$  qui veut minimiser  $F_\alpha$ , i.e qui résout l'équation différentielle

$$w'_\alpha(t) = -\nabla F_\alpha(w_\alpha(t)) = Dh(w_\alpha(t))^T \nabla R(\alpha h(w_\alpha(t)))$$

avec  $w_\alpha(0) = w_0$ . On comparera ce flot de gradient avec celui  $(\bar{w}_\alpha(t))_{t \geq 0}$  de  $\bar{F}_\alpha$  qui résout

$$\bar{w}'_\alpha(t) = -\nabla \bar{F}_\alpha(\bar{w}_\alpha(t)) = Dh(w_0)^T \nabla R(\alpha \bar{h}(\bar{w}_\alpha(t)))$$

avec  $\bar{w}_\alpha(0) = w_0$

### 5.2 Bornes à horizon fini

**Théorème 5.** Si  $h(w_0) = 0$ , alors pour un  $T > 0$ , on a :

- $\sup_{t \in [0, T]} \|w_\alpha(t) - w_0\| = \mathcal{O}(1/\alpha)$
- $\sup_{t \in [0, T]} \|w_\alpha(t) - \bar{w}_\alpha(t)\| = \mathcal{O}(1/\alpha^2)$
- $\sup_{t \in [0, T]} \|\alpha h(w_\alpha(t)) - \alpha \bar{h}(\bar{w}_\alpha(t))\| = \mathcal{O}(1/\alpha)$

### 5.3 Interprétation

Ce qu'il se passe, c'est qu'on a le même genre de résultats que précédemment mais pour n'importe quel modèle lisse : en prenant un paramètre de normalisation  $\alpha$  (par exemple la largeur des couches d'un NN), on se retrouve dans une situation où après un nombre fini de pas de gradients, on est toujours proche de l'initialisation, et encore plus du modèle linéarisé. La sortie régularisée de ce modèle est aussi proche de celle du modèle linéarisé : on se comporte pour  $\alpha$  grand de la même manière qu'un modèle linéaire.

## 6 Conclusion

Dans ce TER, on a présenté le cadre théorique des RKHS, on a mis en évidence des comportements linéaires asymptotiques de fonctions qui ne le sont pas forcément, et on a fait le lien entre la théorie des RKHS et les réseaux de neurones. Il reste plusieurs pistes intéressantes à explorer à présent concernant les réseaux de neurones : avons-nous des résultats concernant la convergence vers des minimums globaux de fonctions de perte ? Si oui avons-nous des résultats sur cette vitesse de convergence ? Ces questions sont répondues dans l'article de M. Chizat [2], et j'aurais volontier traité ces questions avec plus de temps.

## 7 Annexe : Preuves

### 7.1 Théorème : Existence et unicité du noyau reproduisant

Si  $H$  est un RKHS, alors il existe un unique noyau reproduisant de  $H$ .

*Démonstration.* Montrons l'existence puis l'unicité :

- On applique le théorème de Riesz à  $L_x$  :

$$\forall x \in X, \exists! k_x \in H, \forall f \in H, f(x) = L_x(f) = \langle f, k_x \rangle_H$$

Soit  $k(y, x) := k_x(y) \forall x, y \in X$ , alors  $\forall x \in X, f \in H, k(\cdot, x) \in H$  et  $f(x) = \langle f, k(\cdot, x) \rangle_H$ .  
Donc  $k$  est un noyau reproduisant de  $H$ .

- Soient  $k_1$  et  $k_2$  deux noyau reproduisant de  $H$ . Alors  $\forall x \in X, f \in H$

$$\langle f, k_1(\cdot, x) - k_2(\cdot, x) \rangle_H = f(x) - f(x) = 0$$

En prenant  $f = k_1(\cdot, x) - k_2(\cdot, x)$ , on a :

$$\|k_1(\cdot, x) - k_2(\cdot, x)\|_H^2 = 0$$

C'est-à-dire que  $k_1 = k_2$  : le noyau reproduisant de  $H$  est unique. ■

### 7.2 Théorème : Moore-Aronszajn

Soit  $K$  un kernel. Alors il existe un unique espace de Hilbert  $H$  de fonctions sur  $X$  pour lequel  $K$  est un noyau reproduisant.

*Démonstration.*  $\forall x \in X$ , on pose  $K_x(\cdot) := K(x, \cdot)$ . Soit  $H_0$  le sous-espace vectoriel engendré par

$\{K_x : x \in X\}$ . Sous couvert d'existence de  $H$ , la reproducing property de  $K$  nous assure que  $\forall x, y \in X, K(x, y) = \langle K_x, K_y \rangle_H$ . On a de même :

$$\left\langle \sum_{i=1}^n a_i K_{x_i}, \sum_{j=1}^m b_j K_{x_j} \right\rangle_H = \sum_{i=1}^n \sum_{j=1}^m a_i b_j K(x_i, x_j)$$

Il est alors naturel de définir pour toutes fonctions de  $H_0$  :  $f := \sum_{i=1}^n a_i K_{x_i}$  et  $g := \sum_{j=1}^m b_j K_{x_j}$

$$\langle f, g \rangle_{H_0} := \sum_{i=1}^n \sum_{j=1}^m a_i b_j K(x_i, x_j) = \sum_{i=1}^n a_i g(x_i) = \sum_{j=1}^m b_j f(x_j)$$

Les deux dernières égalités nous assure que  $\langle \cdot, \cdot \rangle_{H_0}$  est bien défini. De plus, il est bilinéaire, symétrique et positif, et on a la reproducing property :

$$\forall x \in X, f \in H_0, \langle f, K_x \rangle_{H_0} = \sum_{i=1}^n a_i K_x(x_i) = \sum_{i=1}^n a_i K_{x_i}(x) = f(x)$$

Il reste à montrer que  $\|f\|_{H_0} = 0 \Rightarrow f = 0$  pour que ca soit un produit scalaire : (on utilise CS pour les semi produits scalaires)

$$|f(x)| = |\langle f, K_x \rangle_{H_0}| \leq \|f\|_{H_0} \|K_x\|_{H_0} = \|f\|_{H_0} \sqrt{K(x, x)}$$

Donc  $\langle \cdot, \cdot \rangle_{H_0}$  est un produit scalaire.

On complète  $H_0$  avec le produit scalaire  $\langle \cdot, \cdot \rangle_{H_0}$  en  $H$  (on s'épargnera la longue preuve), et toutes les propriétés sont gardées.

Unicité : supposons  $G$  un autre espace de Hilbert pour lequel  $K$  est un noyau reproduisant.

- $(\subseteq)$  :

On a

$$\forall x, y \in X, \langle K_x, K_y \rangle_H = K(x, y) = \langle K_x, K_y \rangle_G$$

Par linéarité,  $\langle \cdot, \cdot \rangle_H = \langle \cdot, \cdot \rangle_G$  sur tout  $H_0$ , i.e  $H_0 \subseteq G$ . Donc  $H \subseteq G$  car  $G$  est complet.

- $(\supseteq)$  :

Soit  $f \in G$ . Comme  $H$  est un sous-espace fermé de  $G$ , on peut écrire  $f = f_H + f_{H^\perp}$  par le théorème de décomposition orthogonal. De plus comme  $K$  est un noyau reproductif de  $G$  et  $H$  on a  $\forall x \in X$  :

$$f(x) = \langle K_x, f \rangle_G = \langle K_x, f_H \rangle_G + \langle K_x, f_{H^\perp} \rangle_G = \langle K_x, f_H \rangle_G = \langle K_x, f_H \rangle_H = f_H(x)$$

Ainsi  $f \in H$ .

■

### 7.2.1 Representer theorem

La preuve ci-dessous est tirée de [3]. Soit  $k$  un kernel sur  $X$  et soit  $H$  son RKHS associée. Posons  $x_1, \dots, x_n \in X$  notre training sample. Regardons le problème d'optimisation suivant :

$$\min_{f \in H} J(f) := E(f(x_1), \dots, f(x_n)) + P(\|f\|_H^2)$$

Où  $P$  est une fonction croissante.

Alors si ce problème d'optimisation a (au-moins) une solution, il y a (au-moins) une solution de la forme

$$f = \sum_{i=1}^n \alpha_i \cdot k(\cdot, x_i)$$

.

De plus, si  $P$  est strictement croissante, alors toute solution a cette forme.

*Démonstration.* Soit  $H_0$  le sous espace engendré par  $\{k(\cdot, x_i) : i \in 1, \dots, n\}$ . Comme  $H_0 \in H$ ,  $H_0$  est fermé car de dimension finie. Alors le théorème de décomposition orthogonale nous dit que :  $\forall f \in H, f = f_{H_0} + f_{H_0^\perp}$ .

De plus, comme  $k$  est un noyau reproductif de  $H$ , on a  $\forall x_i$  :

$$f(x_i) = \langle k(\cdot, x_i), f \rangle_H = \langle k(\cdot, x_i), f_{H_0} \rangle_H + \langle k(\cdot, x_i), f_{H_0^\perp} \rangle_H = \langle k(\cdot, x_i), f_{H_0} \rangle_H = f_{H_0}(x_i)$$

Alors :

$$\begin{aligned} J(f) &:= E(f(x_1), \dots, f(x_n)) + P(\|f\|_H^2) \\ &= E(f_{H_0}(x_1), \dots, f_{H_0}(x_n)) + P(\|f\|_H^2) \\ &\geq E(f_{H_0}(x_1), \dots, f_{H_0}(x_n)) + P(\|f_{H_0}\|_H^2) \text{ par croissance de } P \text{ car } \|f\|_H^2 = \|f_{H_0}\|_H^2 + \|f_{H_0^\perp}\|_H^2 \\ &= J(f_{H_0}) \end{aligned}$$

Donc si  $f$  est un minimiseur de  $J$ , alors  $f_{H_0} := \sum_{i=1}^n \alpha_i \cdot k(\cdot, x_i)$  aussi. Si  $P$  est strictement croissante, alors l'inégalité est stricte et si l'on veut un minimiseur de  $J$  il faut nécessairement qu'il soit de cette forme. ■

### 7.3 Lois suivant la largeur des couches $m$

- Loi de  $\|u^0\|_2^2$  pour  $m$  grand  
Comme  $\|u^0\|_2^2 = \sum_{i=1}^m (u_i^0)^2 \sim \chi^2(m)$  et  $u_i^0 \sim \chi^2(1)$ , en appliquant le TCL aux  $u_i^0$ , on a :

$$\frac{\|u^0\|_2^2 - m}{\sqrt{2m}} \sim_{m \rightarrow \infty} N(0, 1)$$

En particulier,  $\|u^0\|_2^2 \sim m$  pour  $m$  grand.

- Loi de  $(\frac{1}{m^\gamma} A^0) u^0 x$  sachant  $u^0$

$(A^0 u^0)_i = \sum_{j=1}^m A_{ij}^0 u_j^0$ . En sachant  $u^0$ , comme  $A_{i\cdot}^0$  est un vecteur gaussien,  $(A^0 u^0)_i \sim N(0, \|u^0\|_2^2)$ .

De même, par indépendance des  $A_{ij}^0$ , les  $(A^0 u^0)_i$  sont indépendants et (conditionnellement à  $u^0$ )  $A^0 u^0 \sim N(0_m, \|u^0\|_2^2 Id_m)$ .

Ainsi, la loi de  $(\frac{1}{m^\gamma} A^0) u^0 x$  sachant  $u^0$  est  $N(0_m, (\frac{x}{m^\gamma})^2 \|u^0\|_2^2 Id_m)$ .

- Loi de  $\|(\frac{1}{m^\gamma} A^0) u^0 x\|_2^2$  pour  $m$  grand

On a  $((\frac{1}{m^\gamma} A^0) u^0 x)_i \sim N(0, x^2)$  pour  $m$  grand.

De plus,  $((\frac{1}{m^\gamma} A^0) u^0 x)_i^2 \sim x^2 \cdot \chi^2(1)$ . En appliquant la LGN,  $\|(\frac{1}{m^\gamma} A^0) u^0 x\|_2^2 \sim mx^2$ .

De la même manière que ci-dessus, on peut montrer les deux dernières lois :

- Loi de  $\frac{1}{m^\alpha}(\beta^0)^T x_2$  sachant  $x_2$ , avec  $x_2 = (\frac{1}{m^\gamma} A^0) u^0 x$

On a  $\frac{1}{m^\alpha}(\beta^0)^T x_2 | x_2 \sim N(0, \frac{1}{m^{2\alpha}} \|x_2\|_2^2)$

- Loi de  $\frac{1}{m^\alpha}(\beta^0)^T x_2$  pour  $m$  grand

On a  $\frac{1}{m^\alpha}(\beta^0)^T x_2 \sim N(0, x^2 m^{1-2\alpha})$

#### 7.4 Ordre de grandeur des écarts relatifs

Preuve pour la norme opérateur : le corollaire 7.9 du cours de MIA de M. Giraud [1] nous donne une majoration sur  $|A^0|_{op} : |A^0|_{op} \leq \sqrt{m} + 7\sqrt{m + \xi} = (\sqrt{m})$ , avec  $\xi \sim Exp(1)$ .

De plus, on peut trouver la la norme opérateur de  $\Delta A$  comme suit : tout le travail est de trouver  $|\beta^0(u^0)^T|_{op} := \sup\{|\beta^0(u^0)^T x| \text{ avec } \|x\| = 1\}$ .

$$(\beta^0(u^0)^T)_{ij} = \beta_i^0 u_j^0 \quad (4)$$

$$(\beta^0(u^0)^T x)_i = \sum_{j=1}^m (\beta^0(u^0)^T)_{ij} \cdot x_j \quad (5)$$

$$= \beta_i^0 \langle u^0, x \rangle \quad (6)$$

$$\|\beta^0(u^0)^T x\| = |\langle u^0, x \rangle| \cdot \|\beta^0\| \quad (7)$$

Donc le sup est bien atteint en  $x = u/\|u^0\|$  et est égal à  $\|u^0\| \cdot \|\beta^0\|$ . En utilisant les approximations précédentes, on a donc  $|\beta^0(u^0)^T|_{op} \simeq m$ . Ainsi on a  $|\Delta A|_{op} \sim m^{2\alpha-1} \cdot m^{-1/2-\alpha} \cdot m \sim m^{\alpha-1/2}$ . et :

$$\frac{|\Delta A|_{op}}{|A^0|_{op}} \sim m^{\alpha-1}$$

#### 7.5 Choix de $\alpha$

Preuve de l'ordre de grandeur de  $\|\Delta \nabla_u \Phi\|$  :

$$\Delta \nabla_u \Phi := \nabla_u \Phi(\theta^1, x) - \nabla_u \Phi(\theta^0, x) \quad (8)$$

$$\sim m^{-\alpha-1/2} \underbrace{((\beta^1)^T A^1 - (\beta^0)^T A^0)}_{(\star)} \quad (9)$$

$$(\star) = (\beta^0 - \eta \nabla_\beta F(\Phi(\theta^0, x)))^T (A^0 - \eta \nabla_A F(\Phi(\theta^0, x))) - (\beta^0)^T A^0 \quad (10)$$

$$= \eta^2 [\nabla_\beta F(\Phi)]^T [\nabla_A F(\Phi)] - \eta [\nabla_\beta F(\Phi)] A^0 - \eta \beta^0 [\nabla_A F(\Phi)] \quad (11)$$

$$= (\Delta \beta)^T (\Delta A) + (\Delta \beta) A^0 + \beta^0 (\Delta A) \quad (12)$$

Donc :

$$\|\Delta \nabla_u \Phi\| \sim m^{-\alpha-1/2} \|(\Delta \beta)^T (\Delta A) + (\Delta \beta) A^0 + \beta^0 (\Delta A)\| \quad (13)$$

$$\lesssim m^{-\alpha-1/2} (\|\Delta \beta\| \cdot \|\Delta A\|_{op} + \|\Delta \beta\| \cdot \|A^0\|_{op} + \|\beta^0\| \cdot \|\Delta A\|_{op}) \quad (14)$$

$$\lesssim m^{-\alpha-1/2} (m^{\alpha-1/2} m^{\alpha-1/2} + m^{\alpha-1/2} m^{1/2} + m^{1/2} m^{\alpha-1/2}) \quad (15)$$

$$\lesssim m^{-\alpha-1/2} (m^{\alpha-1/2} m^{1/2} + m^{\alpha-1/2} m^{1/2} + m^{1/2} m^{\alpha-1/2}) \quad (16)$$

$$\lesssim m^{-1/2} \quad (17)$$

C'est-à-dire que  $\|\Delta \nabla_u \Phi\| = \mathcal{O}(m^{-1/2})$  pour  $m$  grand.

Par la même démarche on trouve  $\|\Delta \nabla_\beta \Phi\| = \mathcal{O}(m^{-1/2})$  et  $\|\Delta \nabla_A \Phi\|_F = \mathcal{O}(m^{-1/2})$  pour  $m$  grand.

Preuve que  $F(\Phi(\theta^T, x)) = F(\Phi(\theta^0, x)) + \langle \theta^T - \theta^0, \nabla_\theta F(\Phi(\theta^0, x)) \rangle + \mathcal{O}(m^{-1/2})$  :

En faisant un développement de Taylor, on a :

$$\begin{aligned} F(\Phi(\theta^{t+1}, x)) - F(\Phi(\theta^t, x)) &= \langle \theta^{t+1} - \theta^t, \nabla_\theta F(\Phi(\theta^t, x)) \rangle + \mathcal{O}(m^{-1/2}) \\ \implies \sum_{t=0}^{T-1} F(\Phi(\theta^{t+1}, x)) - F(\Phi(\theta^t, x)) &= \sum_{t=0}^{T-1} \langle \theta^{t+1} - \theta^t, \nabla_\theta F(\Phi(\theta^t, x)) \rangle + \mathcal{O}(m^{-1/2}) \\ \iff F(\Phi(\theta^T, x)) - F(\Phi(\theta^0, x)) &= \sum_{t=0}^{T-1} \langle \theta^{t+1} - \theta^t, \nabla_\theta F(\Phi(\theta^t, x)) \rangle + \mathcal{O}(m^{-1/2}) \\ \iff F(\Phi(\theta^T, x)) &= F(\Phi(\theta^0, x)) + \langle \theta^T - \theta^0, \nabla_\theta F(\Phi(\theta^0, x)) \rangle + \mathcal{O}(m^{-1/2}) \end{aligned}$$

## 7.6 Cas simple

$$\frac{\partial g(u)}{\partial u_j} = \frac{1}{m} \phi'(u_j)$$

$$\|\nabla g(u)\|^2 = \frac{1}{m^2} \sum_{j=1}^m (\phi'(u_j))^2 \quad (18)$$

$$\simeq \frac{1}{m} \mathbb{E}((\phi'(u_j))^2) \text{ pour } m \text{ grand par la LGN} \quad (19)$$

$$\|D^2 g(u)\|_{op} = \sup\{\|D^2 g(u) x\| \text{ avec } \|x\| = 1\}$$

$$(D^2g(u)x)_i = \sum_{j=1}^m (D^2g(u)x)_{ij} x_j \quad (20)$$

$$= \sum_{j=1}^m \frac{\partial g(u)}{\partial u_i \partial u_j} x_j \quad (21)$$

$$= \frac{1}{m} \phi''(u_i) x_i \text{ car la dérivée seconde est nulle si } i \neq j \quad (22)$$

Donc

$$\begin{aligned} \|D^2g(u)x\|^2 &= \frac{1}{m^2} \sum_{j=1}^m x_i^2 (\phi''(u_i))^2 \\ &\leq \frac{1}{m^2} \sup_i (\phi''(u_i))^2 \sum_{j=1}^m x_i^2 \\ &= \frac{1}{m^2} \sup_i (\phi''(u_i))^2 \text{ car } \|x\| = 1 \\ &= \frac{1}{m^2} (\sup_i |\phi''(u_i)|)^2 \\ &= \frac{1}{m^2} (\|\phi''\|_\infty)^2 \end{aligned}$$

Ainsi,

$$\|D^2g(u)\|_{op} \leq \frac{\|\phi''\|_\infty}{m}$$

## 7.7 Dérivées premières

**Propriété 5.** Pour toute matrice réelle  $A$ ,  $\|A\|_{op}^2 = \|A^T A\|_{op}$ , et  $\|A\|_{op} = \|A^T\|_{op}$ .

*Démonstration.* Soit  $v$  le vecteur pour lequel le supremum est atteint dans la définition de  $\|A\|_{op}$  :

$$\begin{aligned} \|A\|_{op}^2 &= \|Av\|^2 = \langle Av, Av \rangle \\ &= \langle A^T Av, v \rangle \\ &\leq \|A^T Av\| \cdot \|v\| \text{ par CS} \\ &= \|A^T A\|_{op} \cdot \|v\| \text{ par définition de } \|\cdot\|_{op} \text{ et car } \|v\| = 1 \\ &= \|A^T A\|_{op} \\ &\leq \|A^T\|_{op} \|A\|_{op} \text{ car } \|\cdot\|_{op} \text{ est sous-multiplicative} \end{aligned}$$

Alors  $\|A\|_{op} \leq \|A^T\|_{op}$ . En substituant  $A^T$  à  $A$ , on a aussi  $\|A^T\|_{op} \leq \|A\|_{op}$  et donc  $\|A^T\|_{op} \|A\|_{op} \leq \|A\|_{op}^2$ . On a encadré  $\|A^T A\|_{op}$  par  $\|A\|_{op}^2$  ce qui conclut la preuve. ■



En appliquant cette propriété,  $\|Dg(u)\|_{op} = \|Dg(u)Dg(u)^T\|_{op}^{1/2}$ .

$$\begin{aligned} Dg(u) &= [D_{u_1}g(u) \quad \cdots \quad D_{u_m}g(u)] \in \mathbb{R}^{n \times (d \cdot m)} \\ &= \frac{1}{m} [D\phi(u_1) \quad \cdots \quad D\phi(u_m)] \end{aligned}$$

$$Dg(u)Dg(u)^T = [D_{u_1}g(u) \quad \cdots \quad D_{u_m}g(u)] \begin{bmatrix} D_{u_1}g(u)^T \\ \vdots \\ D_{u_m}g(u)^T \end{bmatrix} = \frac{1}{m^2} \sum_{j=1}^m D\phi(u_j)D\phi(u_j)^T$$

Asymptotiquement en le nombre  $m$  de couche, on a par la LGN :

$$Dg(u^0)Dg(u^0)^T \simeq \frac{1}{m} \mathbb{E}(D\phi(u_j^0)D\phi(u_j^0)^T)$$

Par continuité de la norme, on a pour  $m$  grand :

$$\|Dg(u)\|_{op} = \|Dg(u)Dg(u)^T\|_{op}^{1/2} \simeq \frac{1}{\sqrt{m}} \|\mathbb{E}(D\phi(u_j^0)D\phi(u_j^0)^T)\|_{op}^{1/2}$$

## 7.8 Dérivées secondes

$$\frac{\partial^2 g(u)}{\partial u_{jk} \partial u_{j'k'}} = \frac{1}{m} \frac{\partial^2 \phi(u_j)}{\partial u_{jk} \partial u_{j'k'}} \mathbf{1}_{j=j'}$$

$D^2g(u)$  est de dimension  $(d \cdot m) \times (d \cdot m)$ , et est une matrice diagonale par blocs de dimension  $d \cdot d$ , dont la diagonale est composée des  $\frac{1}{m} D^2\phi(u_j)$ .

**Propriété 6.** Pour toute matrice réelle symétrique,  $\|A\|_{op} = \sup_{\|x\| \leq 1} |x^T A x|$

*Démonstration.* En diagonalisant  $A$  en  $QDQ^T$  :

•

$$\begin{aligned} \sup_{\|x\| \leq 1} |x^T A x| &= \sup_{\|x\| \leq 1} |x^T QDQ^T x| \\ &= \sup_{\|y\| \leq 1} |y^T D y| \text{ en posant } y := Q^T x \text{ car } Q \text{ est orthogonale.} \\ &= \sup_{\|y\| \leq 1} \left| \sum_k y_k^2 \lambda_k \right| \\ &= |\lambda_{min}| \vee |\lambda_{max}| \text{ en mettant tout le poids sur la valeur propre la plus extrême.} \end{aligned}$$

•

$$\begin{aligned}
\|A\|_{op}^2 &:= \sup_{\|x\| \leq 1} \|Ax\|_2^2 \\
&= \sup_{\|x\| \leq 1} x^T A^T A x \\
&= \sup_{\|y\| \leq 1} y^T D^2 y \\
&= \sup_{\|y\| \leq 1} \left| \sum_k y_k^2 \lambda_k^2 \right| \\
&= (\lambda_{min})^2 \vee (\lambda_{max})^2 \text{ par le même argument que ci-dessus}
\end{aligned}$$

En prenant la racine carrée, on retrouve bien le résultat souhaité. ■

Ainsi :

$$\begin{aligned}
\|D^2 g(u)\|_{op} &= \sup_{\|x\| \leq 1} |x^T D^2 g(u) x| \\
&= \sup_{\sum_{j=1}^m \|x_j\|^2 \leq 1} \frac{1}{m} \left| \sum_{j=1}^m x_j^T D^2 \phi(u_j) x_j \right| \text{ en effectuant un calcul par bloc.} \\
&\leq \frac{1}{m} \sup_{\sum_{j=1}^m \|x_j\|^2 \leq 1} \sum_{j=1}^m |x_j^T D^2 \phi(u_j) x_j| \\
&\leq \frac{1}{m} \sup_{\sum_{j=1}^m \|x_j\|^2 \leq 1} \sum_{j=1}^m |x_j^T D^2 \phi(u_j) x_j| \\
&\leq \frac{1}{m} \sup_{\sum_{j=1}^m \|x_j\|^2 \leq 1} \sum_{j=1}^m \|D^2 \phi(u_j) x_j\| \cdot \|x_j\| \text{ par CS} \\
&\leq \frac{1}{m} \sup_{\sum_{j=1}^m \|x_j\|^2 \leq 1} \sum_{j=1}^m \|D^2 \phi(u_j)\|_{op} \cdot \|x_j\|^2 \text{ par définition de } \|\cdot\|_{op} \\
&\leq \frac{1}{m} \sup_{\sum_{j=1}^m \|x_j\|^2 \leq 1} \max_{j=1 \dots m} \|D^2 \phi(u_j)\|_{op} \sum_{j=1}^m \|x_j\|^2 \\
&\leq \frac{1}{m} \max_{j=1 \dots m} \|D^2 \phi(u_j)\|_{op}
\end{aligned}$$

## 7.9 Bornes à horizon fini : preuve du théorème

**Théorème 6.** Si  $h(w_0) = 0$ , alors pour un  $T > 0$ , on a :

- $\sup_{t \in [0, T]} \|w_\alpha(t) - w_0\| = \mathcal{O}(1/\alpha)$

- $\sup_{t \in [0, T]} \|w_\alpha(t) - \bar{w}_\alpha(t)\| = \mathcal{O}(1/\alpha^2)$
- $\sup_{t \in [0, T]} \|\alpha h(w_\alpha(t)) - \alpha \bar{h}(\bar{w}_\alpha(t))\| = \mathcal{O}(1/\alpha)$

*Démonstration.* Soient  $y(t) := \alpha h(w_\alpha(t))$  et  $\bar{y}(t) := \alpha \bar{h}(\bar{w}_\alpha(t))$ . On a donc

$$\begin{aligned} y'(t) &= \alpha Dh(w_\alpha(t)) \cdot -\frac{1}{\alpha} Dh(w_\alpha(t))^T \nabla R(y(t)) \\ &= -Dh(w_\alpha(t)) Dh(w_\alpha(t))^T \nabla R(y(t)) \\ \bar{y}'(t) &= \alpha Dh(w_0) \cdot -\frac{1}{\alpha} Dh(w_0)^T \nabla R(\bar{y}(t)) \\ &= -Dh(w_0) Dh(w_0)^T \nabla R(\bar{y}(t)) \end{aligned}$$

On pose  $\Sigma(w) := Dh(w) \cdot Dh(w)^T$  pour simplifier les calculs. On a de même  $y(0) = \bar{y}(0) = \alpha h(w_0) = 0$ .

Posons  $C$  une constante, pouvant changer suivant les lignes, indépendante de  $\alpha$ .

$$\int_0^T \|w'_\alpha(t)\| dt = \int_0^T \|\nabla F(w_\alpha(t))\| dt \stackrel{C.S}{\leq} \sqrt{T} \left( \int_0^T \|\nabla F(w_\alpha(t))\|^2 dt \right)^{1/2}$$

Or  $\frac{d}{dt} F(w_\alpha(t)) = \nabla F(w_\alpha(t)) \cdot w'_\alpha(t) = -\|\nabla F(w_\alpha(t))\|^2$ .

On en déduit :

$$\sup_{t \in [0, T]} \|w_\alpha(t) - w_0\| = \left\| \int_0^{t_{sup}} w'_\alpha(t) dt \right\| \leq \int_0^{t_{sup}} \|w'_\alpha(t)\| dt \leq (t_{sup} \cdot F_\alpha(w_\alpha(0)))^{1/2} \lesssim C/\alpha$$

et que  $\sup_{t \in [0, T]} \|y(t) - y(0)\| \leq C$  et  $\sup_{t \in [0, T]} \|\nabla R(y(t))\| \leq C$ . On a retrouvé le premier résultat.

Posons  $\Delta(t) := \|y(t) - \bar{y}(t)\|_{\mathcal{F}}$ . On a  $\Delta(0) = 0$  et  $\Delta'(t) = \langle \frac{y(t) - \bar{y}(t)}{\|y(t) - \bar{y}(t)\|_{\mathcal{F}}}, y'(t) - \bar{y}'(t) \rangle$ . Et par C-S :

$$\begin{aligned} \Delta'(t) &\leq \|\Sigma(w_\alpha(t)) \nabla R(y(t)) - \Sigma(w_0) \nabla R(\bar{y}(t))\|_{\mathcal{F}} \\ &= \|(\Sigma(w_\alpha(t)) - \Sigma(w_0)) \nabla R(y(t)) + \Sigma(w_0) (\nabla R(y(t)) - \nabla R(\bar{y}(t)))\|_{\mathcal{F}} \\ &\stackrel{I.T}{\leq} \|(\Sigma(w_\alpha(t)) - \Sigma(w_0)) \nabla R(y(t))\|_{\mathcal{F}} + \|\Sigma(w_0) (\nabla R(y(t)) - \nabla R(\bar{y}(t)))\|_{\mathcal{F}} \\ &\leq \|\Sigma(w_\alpha(t)) - \Sigma(w_0)\|_{op} \cdot \|\nabla R(y(t))\|_{\mathcal{F}} + \|\Sigma(w_0)\|_{op} \cdot \|\nabla R(y(t)) - \nabla R(\bar{y}(t))\|_{\mathcal{F}} \\ &\leq C_1/\alpha + C_2 \cdot \Delta(t) \end{aligned}$$

Or l'équation différentielle  $u'(t) = C_1/\alpha + C_2 \cdot u(t)$  avec  $u(0) = 0$  a une unique solution qui est  $u(t) = \frac{C_1}{\alpha C_2} (\exp(C_2 t) - 1)$ . Donc d'après le théorème de Petrovitch,  $\Delta(t) \leq \frac{C_1}{\alpha C_2} (\exp(C_2 t) - 1) \leq$

$C/\alpha$ . D'où le troisième résultat.

Maintenant posons  $\delta(t) := \|w_\alpha(t) - \bar{w}_\alpha(t)\|$ .  $\delta'(t) = \langle \frac{w_\alpha(t) - \bar{w}_\alpha(t)}{\|w_\alpha(t) - \bar{w}_\alpha(t)\|}, w'_\alpha(t) - \bar{w}'_\alpha(t) \rangle$ . Et par C-S :

$$\begin{aligned} \delta'(t) &\leq \alpha^{-1} \|Dh(w_\alpha(t))^T \nabla R(y(t)) - Dh(w_0)^T \nabla R(\bar{y}(t))\| \\ &= \alpha^{-1} \|(Dh(w_\alpha(t))^T - Dh(w_0)^T) \nabla R(y(t)) + Dh(w_0)^T (\nabla R(y(t)) - \nabla R(\bar{y}(t)))\| \\ &\leq \alpha^{-1} \|Dh(w_\alpha(t))^T - Dh(w_0)^T\|_{op} \cdot \|\nabla R(y(t))\|_{\mathcal{F}} + \alpha^{-1} \|Dh(w_0)^T\|_{op} \cdot \|\nabla R(y(t)) - \nabla R(\bar{y}(t))\|_{\mathcal{F}} \\ &\leq C/\alpha^2 + C/\alpha^2 \end{aligned}$$

En intégrant par rapport à  $t$ , on retrouve le deuxième résultat. ■

## Références

- [1] Christophe Giraud. *Mathematics for Artificial Intelligence*, page 80. 2020.
- [2] Francis Bach Lénaïc Chizat, Edouard Oyallon. On lazy training in differentiable programming. 2018.
- [3] Clayton Scott. Kernel Methods and the Representer Theorem. [http://web.eecs.umich.edu/~cscott/past\\_courses/eecs598w14/notes/13\\_kernel\\_methods.pdf](http://web.eecs.umich.edu/~cscott/past_courses/eecs598w14/notes/13_kernel_methods.pdf), 2014.