école
normale
supérieure
paris-saclay

Notes de Cours

Fondements Théoriques du deep learning

S.GERCHINOVITZ
F.MALGOUYRES
E.PAUWELS
N.THOME

9 avril 2020

Table des matières

1 Préambule

Ce polycopié contient l'ensemble de mes notes de cours. La majorité des éléments présents dans ce documents ont été présentées en cours ou diffuser numériquement sur le site du cours.
Dans le cadre de ce cours, nous avons abordé plusieurs résultats théoriques sur les réseaux de neurones artificielles. Ces résultats reposent principalement sur des éléments d'algèbre linéaire et de statistique. La quasi totalité des résultats énoncés sont également démontrés avec plus ou moins de détails dans les preuves. Certains théorèmes étant issus d'articles récents, et le temps de cours limité, l'idée était plus d'établir les préliminaires permettant une lecture personnelle plus efficace de ses travaux.
Ce cours a été donné par les enseignants suivant

— Mr.Gerchinovitz : sebastien.gerchinovitz@irt-saintexupery.com

— Mr.Malgouyres : Francois.Malgouyres@math.univ-toulouse.fr

— Mr.Pauwels : edouard.pauwels@irit.fr

— Mr.Thome : nicolas.thome@cnam.fr

Je précise également que je n'ai pas mis l'intégralité des références sur lesquelles s'appuie le cours. De plus, certains cours n'ont pas pu être donnés du au covid-19 et donc la qualité de ma prise de note est moindre sur ces portions.

2 Introduction

On a deux variables aléatoires X et Y définies sur $\mathcal{X}$ et $\mathcal{Y}$. On va observer x une réalisation de X, on va vouloir construire g telle que

$$\begin{array}{rccc} g: & \mathcal{X} & \to & \mathcal{Y} \\ & x & \to & y \end{array}$$

g doit prédire au mieux y, on veut donc trouver

$$g \in \arg\min \mathbb{E}[L(g(X,Y))]$$

On va noter $R(g) = \mathbb{E}[L(g(X,Y))]$ le risque de g qui en effet pour un nouvel échantillon i.i.d le risque observé correspond en effet à cette espérance.

- Dans un problème de **classification** on aura $\mathcal{Y} = \{1, ..., n\}$ pour n classes. En deep learning, on construit une fonction g_c par classe $c \in \mathcal{Y}$ et on définit la prédiction $y = \arg\max g_c(x)$. Pour évaluer l'erreur la méthode usuelle consiste à dénombrer les mauvaises classifications.
- En **régression**, l'espace d'arrivé est euclidien (souvent $\mathbb{R}$ ou $\mathbb{R}^n$) et on prend souvent pour erreur la norme euclidienne au carré de l'écart entre la prédiction et la vérité terrain. On définit la prédiction par coordonnée $y_i = g_i(x)$.

Sous des hypothèses faibles sur la loi de (X,Y) et si elle existe, la fonction définie pour $x \in \mathcal{X}$ par

$$g_b(x) = \arg\min_{y \in \mathcal{Y}} \mathbb{E}[L(y,Y)|X = x]$$

minimise le risque

$$\forall g : \mathcal{X} \to \mathcal{Y} \mathbb{E}[l(g(X),Y)] \geq \mathbb{E}[L(g_b(X),Y)]$$

On appelle g_b la **décision Bayésienne**. On n'a pas toujours de solution telle que $\mathbb{P}(Y = a|X = x) = 1$ et d'ailleurs en général $\mathbb{E}[L(g_b(X),Y)] \neq 0$. Pour montrer ce résultat on écrit

$$\mathbb{E}[L(g_b(X),Y)] = \int_{\mathcal{X}} \int_{\mathcal{Y}} L(g(x),y) d\mathbb{P}_{Y|X}(x,y) d\mathbb{P}_X(x) = \int_{\mathcal{X}} \mathbb{E}[L(g(x),y)|X = x] d\mathbb{P}_X(x)$$

Or par définition ceci est plus grand que $\int_{\mathcal{X}} \mathbb{E}[L(g_b(x),y)|X = x] d\mathbb{P}_X(x) = \mathbb{E}[L(g_b(X),Y)]$. En général, on ne peut pas représenter g_b avec un ordinateur et de toute façon on ne connaît pas la loi de (X,Y). En pratique, on considère un échantillon $(x_i,y_i)_{i \in \{1,...,N\}}$ et une famille de fonction $\mathcal{F}$ de fonctions g_w paramétrées par des paramètres w d'un espace euclidien. Pour nous, ce seront les réseaux de neurones. On cherche à résoudre

$$w^* \in \arg\min_w R(w) = \mathbb{E}[L(g_w(X),Y)]$$

Une approche naïve consiste à essayer de minimiser le risque empirique

$$\hat{w} \in \arg\min_w \hat{R}(w) = \frac{1}{N} \sum_{i=1}^{N} L(g_w(x_i), y_i)$$

Cette minimisation soulève trois questions :

— On s'interroge sur la qualité, en terme d'**optimisation**, de la valeur obtenue $w^{\text{calc}} \sim \hat{w}$ au sens $\hat{R}(w^{\text{calc}}) \simeq \hat{R}(\hat{w})$. En effet, les structures de la fonctions objective ne se prêtent pas forcément à l'optimisation. De plus l'espace est souvent de grande dimension et dépend de beaucoup d'entrées (ceci est à l'origine de l'appellation "Big Data").

— On se demande si la valeur obtenue **généralise** bien, $\hat{w} \sim w^*$ au sens $R(\hat{w}) \simeq R(w^*)$. On veut contrôler $\sup_w \|\hat{R}(w) - R(w)\|$.

— Et enfin se pose la question de l'**approximation**, $g_{w^{\text{calc}}} \sim g_b$ au sens $R(g_{w^{\text{calc}}}) \simeq R(g_b)$. Ce qui peut être difficile à dire si on sait peu de choses sur g_b. On veut contrôler $\inf_w \|R(w) - R(g_b)\|$.

Il s'avère que le terme d'optimisation comme celui d'approximation profite d'une famille de fonctions riche, contrairement au terme de généralisation. En pratique, on observe que le terme de généralisation n'est pas trop gênant. Ce sont les trois principales thématiques de recherche en théorie du deep learning.

2.1 Réseaux de neurones

On représente le réseau comme un graphe avec un ensemble de neurones (nœud) d'entrée et un ensemble de neurone en sortie. Ce découpage en ensemble est un découpage en couche. Chaque arête contient un poids. Un réseau à H couches ou encore $H-1$ couches cachées.

2.1.1 Réseaux feed forward

La structure standard pour les réseaux : **feed forward** (anciennement : **multilayer perceptron**). Chaque couche i est de taille n_i (nombre de neurones).

— On note $f_h(x)$ le **résultat** obtenu en calculant le contenu de la couche pour l'entrée $x \in \mathbb{R}^{n_0}$.

— On note $W_h \in \mathbb{R}^{n_h \times n_{h-1}}$ la **matrice des poids** des arcs entre la couche $h-1$ et la couche h.

— On note $b_h \in \mathbb{R}^{n_h}$ le **biais** ajouté la couche h.

— On note σ la **fonction d'activation** appliquée à chaque couche

On a la récurrence suivante :

$$f_h(x) = \sigma(W_h f_{h-1}(x) + b_h) \quad \text{et} \quad f_0(x) = x$$

On appelle **fully connected** un feed forward si les matrices sont pleines. On dit de plus que le réseau est **linéaire** si on n'a ni biais si fonction d'activation.

Proposition 1. *Pour tout réseau linéaire biaisé feed forward, f_H est affine :*

$$f_H(x) = W_H ... W_1 x + b'_H$$

avec $b'_H = W_H ... W_2 b1 + .. + W_H b_{H-1} + b_H$ obtenu par récurrence

La preuve est immédiate. Une fonction d'activation très utilisée est la **ReLU** (rectified linear unit) définie par

$$\sigma(t) = \max\{t, 0\}, \quad \forall t \in \mathbb{R}$$

Proposition 2. *Pour tout réseau linéaire de fonction d'activation ReLU, on a alors les propriétés suivantes*

- f_H *est continue,*
- f_H *est affine par morceaux,*
- *de au plus* $2^{n_1+...+n_H}$ *morceaux,*
- *chaque morceau est un polyèdre d'au plus* $n_1 + ... + n_H$ *faces.*

(on verra la preuve plus tard) Ce résultat rappelle les cellule de Laguerre/Voronoï (lien ? on a réussi à retrouver les exemples d'entraînement grâce à la connaissance d'un réseau). Pour des fonctions constantes par morceaux on peut utiliser la fonction $\mathbf{1}_{\mathbb{R}_+}$.

2.1.2 Réseaux convolutifs

On applique successivement des filtres via des convolutions à une image et on termine le réseau en feed forward. Le signal x' d'entrée est répété pour donner l'entrée du réseau $x = \begin{pmatrix} x' & ... & x' \end{pmatrix}$. On a des signaux de taille N et on considère H couches avec des tailles comme précédemment. Cependant $n_h = N\times$ nombre de sommets. f_h et W_h correspondent aux mêmes notions que précédemment. Cependant W_h est la concaténation des matrices de convolutions. On retrouve la même formule de récurrence

$$f_h(x) = \sigma(W_h f_{h-1}(x) + b_h) \quad \text{et} \quad f_0(x) = \begin{pmatrix} x & ... & x \end{pmatrix}$$

En somme, les réseaux convolutifs sont des réseaux feed forward particuliers. On suppose $v, x \in \mathbb{N} + \not\Vdash$. Le signal v est $(N+1)$-périodique. On a

$$\begin{pmatrix} v_0 & v_N & ... & v_1 \\ v_1 & v_0 & ... & v2 \\ \vdots & \vdots & \vdots & \vdots \\ v_N & ... & ... & v_0 \end{pmatrix} \times \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^N v_{0-i}x_i \\ \sum_{i=1}^N v_{1-i}x_i \\ \vdots \\ \sum_{i=1}^N v_{N-i}x_i \end{pmatrix}$$

Proposition 3. *Pour tout réseaux convolutif linéaire,* f_H *est affine et le terme linéaire est une convolution du signal de départ.*

$$f_H(x) = W_H...W_1x + b_0H$$

où $b_0H = W_H...W_2b_1 + ... + W_Hb_H - 1 + b_H$.

Ce résultat est le même que pour les réseaux feed forward.

Proposition 4. *Pour tout réseau convolutif de fonction d'activation ReLU, on a alors les propriétés suivantes*

— f_H *est continue,*

— f_H *est affine par morceaux,*

— *de au plus* $2^{n_1+...+n_H}$ *morceaux,*

— *chaque morceau est un polyèdre d'au plus* $n_1 + ... + n_H$ *faces.*

Démonstration. Le point 1 est immédiat puisqu'une composition de fonctions continues est continue. On va montrer les points 2 et 3 en même temps. Pour tout $h \in \{1, ..., H\}$, on note $A_h(x) \in \mathbb{R}^{n_h \times n_h}$ la matrice diagonale telle que $A_h(x)_{i,i} = \mathbf{1}_{[W_h f_{h-1}(x)+b_h]_i \geq 0}$. Ainsi,

$$[A_h(x)(W_h f_{h-1}(x) + b_h)]_i = \sigma(W_h f_{h-1}(x) + b_h)_i = f_h(x)_i$$

On considère l'opérateur $A(x) = (A_h(x))_{1 \leq h \leq H}$ l'activation de ReLU.

$$A : \quad \mathcal{X} \quad \rightarrow \quad \{0,1\}^{n_1 \times n_1} \times ... \times \{0,1\}^{n_H \times n_H}$$

Cet opérateur est à valeur dans un ensemble fini et est donc constant par morceaux. Le nombre de morceaux est alors majoré par le cardinal de l'ensemble $2^{n_1+...+n_H}$. On considère un ensemble $C \subset \mathcal{X}$ un morceau sur lequel A est constant. On a alors pour tout $x \in C$,

$$A(x) = A'$$

On pose pour tout h, $W'_h = W_h b'_h$ et $b'_h = A'_h b_h$ donc pour tout $x \in C$, $\sigma(W_h f_{h-1}(x) + b_h) = W'_h f_{h-1}(x) + b'_h$. Finalement, on a converti notre réseau en réseau linéaire biaisé. Il reste à prouver le point 4. On note $D_h = \{x \in \mathcal{X} | \forall k = 1, ..., h, \quad A(x)_k = A'_k\}$. On a alors par construction que $C = D_h$, on va montrer par récurrence sur h que D_h est un polyèdre d'au plus $n_1 + ... + n_h$ faces. En effet,

$$D_1 = \{x \in \mathcal{X} | (W_1 x + b_1)_i < 0, \forall i \text{ t.q. } (A'_1)_i = 0 \text{ et } (W_1 x + b_1)_i \geq 0, \forall i \text{ t.q. } (A'_1)_i = 1\}$$

Ce qui en fait bien un polyèdre d'au plus n_1 faces. On suppose maintenant que D_{k-1} est un polyèdre à au plus $n_1 + ... + n_{k-1}$ faces. On a alors

$$D_k = \{x \in D_{k-1} | (W_k f_{k-1}(x) + b_k)_i < 0, \forall i \text{ t.q. } (A'_k)_i = 0 \text{ et } (W_k f_{k-1}(x) + b_k)_i \geq 0, \forall i \text{ t.q. } (A'_k)_i = 1\}\}$$

□

Notons que les morceaux ne sont pas nécessairement connexes. Notons également que cette preuve ne tient pas compte de la structure de W_h et peut donc s'appliquer aux réseaux feed forward. De même remarquons que même sur un morceau, la partie linéaire de f_H contient l'action de la fonction d'activation. Elle n'est plus invariante par translation. Ce n'est plus une convolution. Or les contraintes sont de nouveaux affines et ainsi, D_k est un polyèdre d'au plus $n_1 + ... + n_k$ faces. Ceci achève la preuve.

2.1.3 Réseaux structurés

Ce formalisme permet de définir d'une façon les réseaux feed forward ainsi que les réseaux convolutifs et la plus part des réseaux que l'on va rencontrer. On considère des opérateurs linéaires

$$\begin{array}{rccc} W_h: & \mathbb{R}^S & \to & \mathbb{R}^{n_h \times n_{h-1}} \\ & w & \to & W_h(w) \end{array}$$

Chaque couche est alors obtenue par récurrence

$$f_0(x) = x \quad \text{et} \quad f_h(x) = \sigma(W_h(w_h) f_{h-1}(x) + b_h)$$

On peut intégrer la répétition de x dans W_1 pour obtenir un réseau convolutif et on aurait alors pour tout h, W_h concatène des matrices de convolutions. De plus, W_h peut aussi contenir l'action de ReLU et ainsi tout réseau ReLU est un réseau linéaire structuré par morceaux.

2.1.4 Autres réseaux

Notons que nous n'avons pas tout présenté : dont les couches res-net

$$f_h(x) = \sigma(W_h f_{h-1}(x) + b_h + f_{h-2}(x))$$

ni des réseaux récursifs et des réseaux récurrents. On a également les combinaisons de réseaux :

- **Autoencoder** (VAE) : Un réseau compresse, un réseau décompresse
- **Création d'embeddings** : Deux réseaux envoient une image et un texte dans un même espace de caractéristiques. (Visual Query Answering (VQA))
- **Generative Adversarial Networks** (GAN) : un réseau génère des données ; un réseau discrimine les données générées de vraies données.
- **Réduction de biais dans les données** (FairGAN) : Un réseaux envoie les données dans un espace de caractéristiques et ses poids sont optimisés pour qu'un réseau classifiant sur un critère pertinent fonctionne, un réseau classifiant sur un critère non-pertinent échoue.

2.2 Le contrôle du risque

En classification et en régression on cherche à contrôler le risque

$$R(\tilde{f}_w)\mathbb{E}[L(\tilde{f}_w(X), Y)]$$

avec $f_w(X)$ la prédiction et

- en classification : $\tilde{f}_w(x) = f_w(x)$
- en régression : $\tilde{f}_w(x) \in \arg\max(f_w(x))_i$

En effet la personne $\tilde{f}_w$ observe

$$R_{test}(\tilde{f}_w) = \frac{1}{n'} \sum_{i=1}^{n'} L(\tilde{f}_w(x'_i), y'_i)$$

pour un échantillon i.i.d. La loi des grands nombres conduit donc à considérer R. On appelle risque empirique $\hat{R}$ et risque R de g

$$\hat{R}(g) = \frac{1}{n'} \sum_{i=1}^{n'} L(g(x_i), y_i) \text{ et } R(g)\mathbb{E}[L(g(X), Y)]$$

On rappelle la définition du risque bayésien $R^* = \arg\min_g R(g)$. Pour $\epsilon > 0$, on introduit la notion d'ϵ-minimiseur, on fixe w^* et $\hat{w}$ tels que

$$R(\tilde{f}_{w^*}) \leq \min_w R(f_w) + \epsilon \text{ et } \hat{R}(\tilde{f}_{\hat{w}}) \leq \min_w \hat{R}(\tilde{f}_w) + \epsilon$$

Pour un $w$retourné par un algorithme, on décompose l'excès de risque selon Fig 1.

$0 \leq$	$R(\tilde{f}_{\mathbf{w}})$	$-$	R^*	$=$	(excès de risque)
	$R(\tilde{f}_{\mathbf{w}^*})$	$-$	R^*	$+$	(erreur d'approximation)
	$\hat{R}(\tilde{f}_{\mathbf{w}^*})$	$-$	$R(\tilde{f}_{\mathbf{w}^*})$	$+$	(erreur de généralisation)
	$\hat{R}(\tilde{f}_{\mathbf{w}'})$	$-$	$\hat{R}(\tilde{f}_{\mathbf{w}^*})$	$+$	$\leq \varepsilon$
	$\hat{R}(\tilde{f}_{\mathbf{w}})$	$-$	$\hat{R}(\tilde{f}_{\mathbf{w}'})$	$+$	(erreur d'optimisation)
	$R(\tilde{f}_{\mathbf{w}})$	$-$	$\hat{R}(\tilde{f}_{\mathbf{w}})$		(erreur de généralisation)

FIGURE 1 – Décomposition du risque

— **Erreur d'approximation** : il suffit que $\|\tilde{f}_{w^*} - g_b\|$ soit petit. Quelles fonctions peut-on approximer avec quels réseaux ? ("Expressive power", "Expressivité",...)

— **Erreur de généralisation** : Quelle condition sur le réseau pour que $\hat{R}(\tilde{f}_w) \simeq R(\tilde{f}_w)$, i.e. $\sup_w |\hat{R}(\tilde{f}_w) - R(\tilde{f}_w)|$ pour tout w ? Combien d'échantillons faut-il ? (*Dimension de Vapnik-Chervonenkis*, *Complexité de Rademacher...*)

— **Erreur d'optimisation** : Quand est-ce-que l'optimisation marche ? (Optimisation non-convexe, paysage de la fonction objectif...)

Pour un w obtenu via un algorithme, on a

$$R(\tilde{f}_w) = R(\tilde{f}_w) - \hat{R}(\tilde{f}_w) + \hat{R}(\tilde{f}_w)$$

où $R(\tilde{f}_w) - \hat{R}(\tilde{f}_w)$ est l'erreur de généralisation et $\hat{R}(\tilde{f}_w)$ est observable. Ceci est important que dans le cas où $\hat{R}(\tilde{f}_w)$ est petit, i.e. on a résolu les problème de l'expressivité et de l'optimisation. On parle de **stabilité** du minimiseur si et seulement si on une condition

sur le réseau et les échantillons garantissant une propriété de stabilité du type : Il existe C telle que pour ϵ petit et pour tout w et w^* tels que

$$\hat{R}(\tilde{f}_w) \leq \epsilon \text{ et } \hat{R}(\tilde{f}_{w^*}) \leq \epsilon$$

on ait

$$d(w, w^*) \leq C\epsilon$$

pour une métrique d. Ceci vient du domaine du compressed sensing. On contrôle donc le risque si

— La condition sur le réseau est satisfaite,

— L'optimisation marche et permet de trouver w,

— Hypothèse d'un modèle génératif : Il existe w^* compatible avec les données, pour lequel $R(\tilde{f}_{w^*})$ et $\hat{R}(\tilde{f}_{w^*})$ sont petits.

2.3 Les réseaux comme une fonction des paramètres

2.3.1 Structure de la fonction objectif

On se place dans le même contexte que précédemment. Et on pose

$$\begin{array}{rccc} E: & \mathbb{R}^{H\times S} \times \mathbb{R}^{n_1+\ldots+n_H} & \to & \mathbb{R} \\ & (w,b) & \to & \sum_{i=1}^n L(\tilde{f}_{w,b}(x_i), y_i) \end{array}$$

Proposition 5. *Pour tout réseau, avec la fonction d'activation ReLU, pour tout échantillon d'apprentissage, la fonction E n'est pas coercive.*

Démonstration. L'argument de la preuve est un argument d'homogénéité, en effet on considère un certain $w \in \mathbb{R}^{H\times S}$ et on définit w^t comme suit

$$w_i^t = t^{H-1} w_1 \text{ et } w_h^t = t^{-1} w_h$$

On a alors pour tout t

$$f_{w,0}(x) = f_{w^t,0}(x)$$

Donc E n'est pas coercive □

On a également

Proposition 6. *Pour tout réseau linéaire structuré (i.e. $\sigma = Id$), pour tout $x \in \mathbb{R}^{n_0}$, pour tout $j \in \{1, ..., n_H\}$ la fonction*

$$\begin{array}{rccc} & \mathbb{R}^{H\times S} \times \mathbb{R}^{n_1+\ldots+n_H} & \to & \mathbb{R} \\ & (w,b) & \to & |f_{w,b}(x)| \end{array}$$

est un polynôme de degré H.

Ainsi, si l est quadratique, alors E est un polynôme de degré $2H$. Ceci découle de la définition du réseaux linéaire structuré. Dans le cas de la ReLu on a

Proposition 7. *Pour tout réseau linéaire avec σ =ReLU, pour tout $x \in \mathbb{R}^{n_0}$, pour tout $j \in \{1, ..., n_H\}$ la fonction*

$$\begin{array}{rcl} \mathbb{R}^{H\times S} \times \mathbb{R}^{n_1+...+n_H} & \to & \mathbb{R} \\ (w, b) & \to & |f_{w,b}(x)| \end{array}$$

est continue et est un polynôme de degré H par morceaux.

Comme vu précédemment, il y a au plus $2^{n_1+...+n_H}$ morceaux. Et encore une fois, si l est quadratique, alors E est un polynôme de degré $2H$ par morceaux. Notons de plus que les bords des morceaux sont de mesures nulles.
Nous allons prouver la dernière proposition. Cette preuve ressemble à une preuve précédente.

Démonstration. La fonction $(w, b) \to [f_{w,b}(x)]_i$ est clairement continue. Pour tout h dans $\{1, ..., H\}$, on note $B_h(w, b) \in \mathbb{R}^{n_h\times n_h}$ la matrice diagonale telle que pour tout i dans $\{1, ..., n_h\}$ on a

$$[B_h(w, b)]_i = \mathbf{1}_{[W_h(w_h)f_{h-1}(x)+b_h]_i\geq 0}$$

On a donc

$$[B_h(w, b)|W_h(w_h)f_{h-1}(x) + b_h]_i = [\sigma(W_h(w_h)f_{h-1}(x) + b_h)]_i = [f_h]_i$$

On note $B(w, b) = (B_h(w, b))_{1\leq h\leq H}$. Cette fonction est à valeurs dans un ensemble fini, B est donc constante par morceaux, et il y a au plus $2^{n_1+...+n_H}$ morceaux. Soit $C \subset \mathbb{R}^{H\times S} \times \mathbb{R}^{n_1+...+n_H}$ un morceau sur lequel B est constante. On note B' cette constante, $W'_h(w_h) = B'_h W_h(w_h)$ et $b'_h = B'_h b_h$ pour tout (w, b) dans C. On a alors pour tout h

$$f_h(x) = W'_h(w_h)f_{h-1}(x) + b'_h$$

Ainsi l'action d'un réseau ReLU coïncide avec l'action d'un réseau linéaire structuré défini par W' et b' sur C. On peut donc appliquer le résultat précédent et conclure. □

Comme précédemment, les morceaux ne sont pas nécessairement connexes. Encore aujourd'hui on ne sait pas grand chose sur les morceaux. Les inégalités polynomiales sont étudiées en géométrie algébrique. Les résultats précédents de contrôle du risque issus du compressed sensing, exploite le fait que les polynômes des propositions précédentes ne peuvent être quelconques (on aura pas de terme en w_1^2).

2.4 La Back-propagation

En français rétro-propagation, on considère toujours le coût

$$\begin{array}{rcl} E: \ \mathbb{R}^{H\times S} \times \mathbb{R}^{n_1+...+n_H} & \to & \mathbb{R} \\ (w, b) & \to & \sum_{i=1}^n L(\tilde{f}_{w,b}(x_i) - y_i) \end{array}$$

Dans cette partie, nous allons définir le point clef de l'algorithme de descente de gradient stochastique dans le cas des réseaux de neurones. On considère un unique exemple (x, y) (si besoin on somme les gradients). On traite le cas où L est C^1. On suppose que σ est C^1 (on peut régulariser σ). On note l'action de la couche h

$$\begin{array}{rccc} c^h_{w,b}: & \mathbb{R}^{n_{h-1}} & \to & \mathbb{R}^n_h \\ & f & \to & \sigma(W_h(w_h)f + b_h) \end{array}$$

On note également $F^h_{w,b}$ qui vaut l'identité pour $h = H$ et et sinon correspond à la composition des actions des couches restantes et $f^{h-1}_{w,b}$ la réciproque (identité pour $h = 0$ et composition des actions précédentes à h). Ainsi

$$f_{w,b} = F^h_{w,b} \circ c^h_{w,b} \circ f^{h-1}_{w,b} = F^h_{w,b} \circ f^h_{w,b}$$

et par récurrence on a $F^h_{w,b} = F^{h+1}_{w,b} \circ c^{h+1}_{w,b}$

Proposition 8. *Si w, b, x sont tels que $f_{.,.}(x)$ soit différentiable, alors*

$$\nabla E(w, b) = (J_{f_{.,.}(x)}(w, b))^T . \nabla l(f_{w,b}(x) - y)$$

Démonstration. On a

$$E((w, b) + (w', b')) = l(f_{w+w',b+b'}(x) - y)$$

On fait un développement limité de $f_{.,.}(x)$

$$E((w, b) + (w', b')) = l\left(f_{w,b}(x) - y + J_{f_{.,.}(x)}(w, b)\begin{pmatrix} w' \\ b' \end{pmatrix} + o(\|(w', b'\|))\right)$$

On fait maintenant un développement limité de l

$$E((w, b) + (w', b')) = E(w, b) + \left\langle \nabla l(f_{w,b}(x) - y); J_{f_{.,.}(x)}(w, b)\begin{pmatrix} w' \\ b' \end{pmatrix} \right\rangle + o(\|(w', b'\|)$$

On obtient ainsi

$$E((w, b) + (w', b')) = E(w, b) + \left\langle (J_{f_{.,.}(x)}(w, b))^T . \nabla l(f_{w,b}(x) - y); \begin{pmatrix} w' \\ b' \end{pmatrix} \right\rangle + o(\|(w', b'\|)$$

Ce qui achève la preuve. □

On doit encore savoir comment calculer $J_{f_{.,.}(x)}(w, b)$. On note la dérivée $d^h_{w,b}$

$$\begin{array}{rccc} d^h_{w,b}: & \mathbb{R}^{n_{h-1}} & \to & \mathbb{R}^{n_h} \\ & f & \to & \sigma'(W_h(w_h)f + b_h) \end{array}$$

Proposition 9. *Si w, b, x sont tels que $f_{.,.}(x)$ soit différentiable, alors*

$$J_{f_{.,.}(x)}(w, b).\begin{pmatrix} w' \\ b' \end{pmatrix} = \sum_{h=1}^{H} J_{F^h_{w,b}(.)}(f^h_{w,b}(x)).diag(d^h_{w,b} f^{h-1}_{w,b}(x)).(W_h(w_h) f^{h-1}_{w,b}(x) + b'_h)$$

Démonstration. On a bien

$$J_{f_{.,.}(x)}(w',b').\begin{pmatrix}w'\\b'\end{pmatrix}=\sum_{h=1}^{H}J_{f_{.,.}(x)}(w,b)\begin{pmatrix}w'_h\\b'_h\end{pmatrix}$$

On a également par développement limité de σ

$$c^h_{(w,b)+(w',b')}(f)=c^h_{(w,b)}(f)+\text{diag}(\sigma'(W_h(w_h)f+b_h)).(W_h(w'_h)f+bi_h)+o(\|(w',b'\|)$$

Donc,

$$f_{(w,b)+(w',b')}(x)=F^h_{w,b}(c^h_{(w,b)+(w',b')}(f^{h-1}_{w,b}(x)))$$

i.e.

$$f_{(w,b)+(w',b')}(x)=F^h_{w,b}(f^h_{w,b}(x)+D.(W_h(w'_h)f^{h-1}_{w,b}(x)+bi_h)+o(\|(w',b'\|))$$

où $D=\text{diag}(\sigma'(W_h(w_h)f^{h-1}_{w,b}(x)+b_h))$ Enfin on fait un développement limité de $F^h_{w,b}$ et on obtient

$$f_{(w,b)+(w',b')}(x)=f_{(w,b)}(x)+J_{F^h_{w,b}(.)}(f^h_{w,b}(x)).D.(W_h(w_h)f^{h-1}_{w,b}(x)+b'_h)+o(\|(w',b'\|))$$

□

Il reste à calculer $J_{F^h_{w,b}(.)}(f)$.

Proposition 10. *Si w,b,x sont tels que $F^h_{w,b}(.)$ soit différentiable, alors*

$$J_{F^h_{w,b}(.)}(f)=\begin{cases}Id_{n_H} & si\ h=H\\ J_{F^{h+1}_{w,b}(.)}(c^{h+1}_{w,b}(f)).diag(d^{h+1}_{w,b}(f)).W_{h+1}(w_{h+1}) & sinon\end{cases}$$

C'est cette récurrence qui définit la back-propagation.

Démonstration. Si $h=H$, on a $F^H_{w,b}(f)=f$ et donc clairement

$$J_{F^h_{w,b}(.)}(f)=\text{Id}_{n_H}$$

Si on considère maintenant $h<H$, et f,f' dans $\mathbb{R}^{n_h}$. On a alors

$$F^h_{w,b}(f+f')=F^{h+1}_{w,b}(c^{h+1}_{w,b}(f+f'))$$

On développe

$$F^h_{w,b}(f+f')=F^{h+1}_{w,b}(\sigma(W_{h+1}(w_{h+1})(f+f')+b_{h+1}))$$

On distribue et on fait le développement limité de σ

$$F^h_{w,b}(f+f')=F^{h+1}_{w,b}\big(\sigma(W_{h+1}(w_{h+1})f+b_{h+1}).D.W_{h+1}(w_{h+1})f'+o(\|f'\|)\big)$$

avec $D=\text{diag}(\sigma'(W_{h+1}(w_{h+1})f+b_{h+1}))$. On fait le développement limité de F

$$F^h_{w,b}(f+f')=F^h_{w,b}(f)+J_{F^{h+1}_{w,b}(.)}(c^{h+1}_{w,b}(f)).\text{diag}(d^{h+1}_{w,b}(f)).W_{h+1}(w_{h+1})+o(\|f'\|)$$

□

2.4.1 Cas particulier : réseau feed-forward

Supposons que nous sommes dans le cas d'un réseau feed-forward fully-connected. On note

$$\begin{array}{rccc} M_h: & \mathbb{R}^S & \to & \mathbb{R}^{n_h \times n_{h-1}} \\ & w & \to & (w_{(i-1)n_{h-1}+j})_{1\leq i\leq n_h, 1\leq j\leq n_{h-1}} \end{array}$$

Proposition 11. *En posant $k = (i-1)n_{h-1} + j$ on a*

$$\frac{\partial E}{\partial w_{h,k}}(w,b) = [f_{w,b}^{h-1}(x)]_j [d_{w,b}^h(f_{w,b}^{h-1}(x))]_i \Delta_i^h(x)$$

et

$$\frac{\partial E}{\partial b_{h,i}}(w,b) = [d_{w,b}^h(f_{w,b}^{h-1}(x))]_i \Delta_i^h(x)$$

avec

$$\Delta^h(x) = \begin{cases} \nabla l(f_{w,b}(x) - y) & si\ h = H \\ [W_{h+1}(w_{h+1})]^T . diag(d_{w,b}^{h+1}(f_{w,b}^h(x))) . \Delta^{h+1} & sinon \end{cases}$$

Démonstration. On note $\mathrm{lig}(h,k)$ et $\mathrm{col}(h,k)$ la ligne et la colonne correspondant à $w_{h,k}$ On a

$$\frac{\partial E}{\partial w_{h,k}}(w,b) = \nabla E(w,b)_{\mathrm{lig}(h,k)}$$

En utilisant la proposition 9,

$$J_{f_{.,.}(w)}(w,b)_{\mathrm{col}(h,k)} = J_{F_{w,b}^h(.)}(f_{w,b}^h(x)).\mathrm{diag}(d_{w,b}^h f_{w,b}^{h-1}(x)).(W_h(w_h) f_{w,b}^{h-1}(x) + b_h')\delta_i$$

Donc en utilisant la proposition 8, on a

$$\frac{\partial E}{\partial w_{h,k}}(w,b) = \langle (J_{f_{.,.}(x)}(w,b)); \nabla l(f_{w,b}(x) - y) \rangle$$

i.e.

$$\frac{\partial E}{\partial w_{h,k}}(w,b) = \langle [f_{w,b}^{h-1}(x)]_j [d_{w,b}^h f_{w,b}^{h-1}(x)]_i J_{F_{w,b}^h(.)}(f_{w,b}^h(x)\delta_i; \nabla l(f_{w,b}(x) - y) \rangle$$

i.e.

$$\frac{\partial E}{\partial w_{h,k}}(w,b) = [f_{w,b}^{h-1}(x)]_j [d_{w,b}^h(f_{w,b}^{h-1}(x))]_i [\Delta^h(x)]_i$$

On fait de même pour $\frac{\partial E}{\partial b_{h,i}}(w,b)$. □

Ceci conclut notre introduction.

3 Optimisation non-Convexe : Implications en Deep Learning

En reprenant les notations précédentes, on s'intéresse au problème d'optimisation suivant

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^{n} L(f_{w,b}(x_i), y_i)$$

On a un échantillon d'entraînement $(x_i, y_i)_{i\in\{1,\dots,n\}}$. On considère $f_{w,b} : \mathcal{X} \to \mathcal{Y}$ est un réseau de neurone. Par soucis de simplification on va noter θ les paramètres du réseau et on va noter $l_i(\theta) = L(f_{w,b}(x_i), y_i)$, ainsi on se retrouve avec le problème suivant

$$\min_{\theta} F(\theta) = \frac{1}{n} \sum_{i=1}^{n} l_i(\theta)$$

Calculer un sous-gradient stochastique (cette notion est définie plus loin) de F est comparable à l'évaluation de la fonction en terme de coût de calcul. Cette fonction est à priori non-lisse et non-convexe. Ce genre de problème est à la jonction des thèmes suivants :

— Systèmes dynamiques lisses : Poincaré, Hadamard, Lyapounov, Hirsch,...

— Structure Géométrique favorable de F Lojasiewcz, Hironaka, Grothendiek,...

On va aborder les résultats suivants

— convergence vers un point critique du second ordre pour les fonctions de Morses.

— Structure favorable des objets étudiés en deep learning

— Hypothèses de convergence de Lojasiewcz

— Approche vers un point critique avec la méthode de descente de gradient stochastique.

3.1 convergence vers un minimum local : les fonctions de Morses

On considère le système dynamique suivant

$$\begin{array}{cc} x = S(x) & \text{(flow)} \\ x_{k+1} = T(x_k) & \text{(discret)} \end{array}$$

Avec $S, T : \mathbb{R}^p \to \mathbb{R}^p$ sont deux difféomorphismes locaux. On s'intéresse donc au comportement en temps long. On constate que les systèmes non-linéaires se comportent essentiellement comme leurs approximations linéaires.

Proposition 12. *Soit F une fonction C^2, si le gradient de F est L-Lipschitz, alors la fonction*

$$T : x \to x - \alpha \nabla F(x)$$

est un difféomorphisme pour $0 < \alpha \frac{1}{L}$

Démonstration. Pour tout $x \in \mathbb{R}^p$, la jacobienne $\nabla T = I - \alpha\nabla^2 F$ est définie positive et donc inversible localement. En conséquence du théorème des fonctions implicites on déduit que T est un difféomorphisme local. De plus par Lipschitzianité

$$\|x - y\| = \alpha\|\nabla F(x) - \nabla F(y)\| < L\alpha\|x - y\|$$

Et donc $x = y$. Ce qui donné que T est bien un difféomorphisme. l'inverse explicite est solution du problème convexe

$$\arg\min_{y\in\mathbb{R}^p} -\alpha F(y) + \frac{1}{2}\|y - z\|_2^2$$

□

On se demande alors ce qu'il se passe en temps grand avec $x_{k+1} = ax_k$, $a \in \mathbb{C}$

— $\|\alpha\| < 1 : x_k \to 0$
— $\|\alpha\| = 1 : x_k \in \mathcal{S}$ avec $\mathcal{S}$ une sphère
— $\|\alpha\| > 1$: pour $x_0 \neq 0$, (x_k) diverge

Si maintenant $x_{k+1} = Dx_k$, avec D une matrice diagonale. On retrouve le résultat précédent coordonnées par coordonnées. Si on a une matrice M symétrique réelle sans valeur propre de module égal à 1 alors on a

$$\mathbb{R}^p = E_s \oplus E_u$$

Avec E_s l'espace stable de M qui est l'espace propre des valeurs propres de module inférieur à 1. De plus si $dim(E_u) > 0$ alors il existe un comportement divergent générique presque sûrement, i.e.

$$\mathbb{P}((x_k) \text{ converge}) = \mathbb{P}(x_0 \in E_s) = \lambda(E_s) = 0$$

théorème 13. *Soit $T : \mathbb{R}^p \to \mathbb{R}^p$ un difféomorphisme local en $\bar{x}$ un point fixe de T tel que les valeur propres de ∇T n'appartiennent pas à la sphère unité et tel qu'au moins une valeur est hors de la boule unité. Alors il existe un voisinage U de $\bar{x}$ tel que*

$$W^s(T, \bar{x}) = \{x_0 \in U | T^n(x_0) \to \bar{x}, n \to \infty\}$$

et

$$W^u(T, \bar{x}) = \{x_0 \in U | T^n(x_0) \to \bar{x}, n \to -\infty\}$$

sont des variétés tangentes différentiables aux espaces stables et instables de $\nabla T(\bar{x})$. En particulier $dim(W^s) < p$.

On suppose que F est C^2, toujours avec un gradient L-Lipschitz avec $\bar{x}$ vérifiant

$$\nabla F(\bar{x}) = 0$$
$$\nabla^2 F(\bar{x}) \text{ n'a aucune valeur propre nulle}$$
$$\nabla^2 F(\bar{x}) \text{ a au moins une valeur propre négative}$$

Une fonction vérifiant les deux premières propriétés sont appelées **fonctions de Morse**.

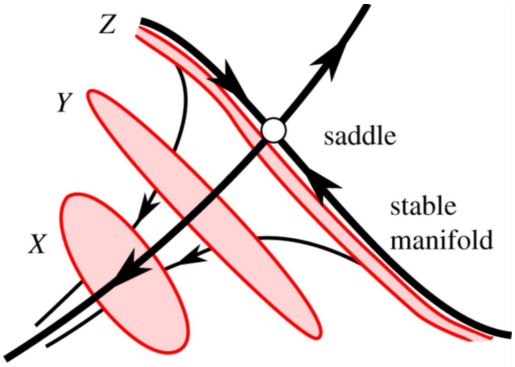

FIGURE 2 – Ici la variété stable est représentée par les flèches allant vers le point selle et la variété instable est représentée par les flèches fuyant le point selle

Proposition 14. *On suppose x_0 choisit au hasard. Si on considère la suite (x_k) partant de x_0 avec $\alpha < 1/L$ alors*

$$\mathbb{P}(x_k \to \bar{x}) = 0$$

Démonstration. La fonction $T : x \to x - \alpha \nabla F(x)$ satisfait les conditions du théorème précédent. Ceci implique que si on a $x_k \to \bar{x}$ alors il existe un k^* tel que pour tout $k > k^*$ on a $x_k \in W^s$ et donc

$$W^s = \cup_{k^* \in \mathbb{N}} T^{-k^*}(W^s(T, \bar{x}))$$

Or T est un difféomorphisme, donc

$$\lambda(W^s(T, \bar{x})) = 0 \Leftrightarrow \lambda(\cup_{k^* \in \mathbb{N}} T^{-k^*}(W^s(T, \bar{x}))) = 0$$

□

3.2 Structure Favorable des objets étudiés en Deep Learning

On suppose maintenant que la fonction de coût est soit du type L^1 ou L^2 et que le réseau ne contient que des activations ReLU. On déduit du chapitre précédent que F est polynomiale par morceaux.

$$\begin{array}{rccc} F: & \mathbb{R}^p & \to & \mathbb{R} \\ & \theta & \to & \frac{1}{n}\sum_{i=1}^n l_i(\theta) \end{array}$$

Un ensemble **semi-algébrique** de $\mathbb{R}^p$ est une union finie de solutions de systèmes de la forme

$$\{x \in \mathbb{R}^p | P(x) = 0, Q_1(x) > 0, ..., Q_l(x) > 0\}$$

Où $P, Q_1, ..., Q_l$ sont des polynômes. Un ensemble est dit algébrique lorsque les conditions sont à égalités. Une fonction est dite semi-algébrique si et seulement si son graphe est semi-algébrique. Dans le cas $p = 1$, un ensemble semi-algébrique est une union finie

d'intervalles. F est semi-algébrique comme toute fonction polynomiale par morceaux. Par exemple,

$$z = ReLU(x) \leftrightarrow z(z - x) = 0, z \geq 0, z \geq x$$

théorème 15. Tarski-Seidenberg *Soient $A \subset \mathbb{R}^{p+1}$ un ensemble semi-algébrique et π une projection sur les p premières coordonnées, alors $\pi(A)$ est un ensemble semi-algébrique.*

(Ce théorème est difficile) En conséquence, tout ensemble ou fonction pouvant être décrit en utilisant une formule du premier ordre (quantification sur les variables et donc pas sur les ensembles) avec des variables réelles, des objets semi-algébriques, des additions, des multiplications et des égalités/inégalités est semi-algébrique. Par exemple

— L'image ou la pré-image d'une fonction semi-algébrique,

— L'intérieur, l'adhérence ou le bord d'un ensemble semi-algébrique,

— La dérivée d'une fonction semi-algébrique dérivable,

— L'ensemble des discontinuités et singularités d'une fonction semi-algébrique.

(Pour plus de détails voir : Michel Coste's Introduction to semi-algebraic geometry [5])

théorème 16. Morse-Sard *Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction semi-algébrique différentiable alors l'ensemble des points critiques de f est fini*

$$crit_f = f(\{x \in \mathbb{R} | f'(x) = 0\})$$

Démonstration. On pose $C = \{x \in \mathbb{R} | f'(x) = 0\}$ est semi-algébrique car f' est semi-algébrique. On peut alors écrire C comme union finie d'intervalles, et pour tout a, b dans un de ces intervalles, on a

$$f(b) - f(a) = \int_a^b f(t')dt = 0$$

Ainsi f est constante sur chaque intervalle et prend donc un nombre fini de valeurs. □

3.3 Structure o-minimale

Une **structure o-minimale** est obtenue avec une définition axiomatique. On a $\mathcal{M} = \cup_{p\in\mathbb{N}}\mathcal{M}_p$ avec $\mathcal{M}_p$ un sous-ensemble de $\mathbb{R}^p$ vérifiant

— stabilité par union, intersection et passage au complémentaire

— si $A \in \mathcal{M}_p$ et $B \in \mathcal{M}_{p'}$ alors $A \times B \in \mathcal{M}_{p+p'}$

— chaque $\mathcal{M}_p$ contient les ensembles semi-algébrique de $\mathbb{R}^p$

— si $A \in \mathcal{M}_{p+1}$ alors $\pi(A) \in \mathcal{M}_p$

— $\mathcal{M}_1$ consiste en l'ensemble des unions finies d'intervalles

Une fonction est dire **modérée** si et seulement si son graphe appartient à une structure o-minimale. On a quelques exemples de structures o-minimales

— ensembles semi-algébriques (Tarski-Seidenberg)

— ensembles exponentiels (Wilkie)

— restriction de fonctions analytiques à des ensembles bornés (Gabrielov)

l'axiome 4 permet de faire de l'élimination de quantificateur. Ainsi F est semi-algébrique pour toute activation/perte semi-algébrique. Pour la plupart des choix de d'activation/-perte F est modérée.

3.4 Convergence des fonctions modérées

On considère que F est C^1 et de gradient L-Lipschitz sans se préoccuper du lien avec les réseaux de neurones pour le moment. On prend $\alpha \in]0, 1/L]$, ainsi

$$x_{k+1} = x_k - \alpha \nabla F(x_k)$$

On introduit les fonctions **dé-singularisante**, pour éviter les comportement spiralant,

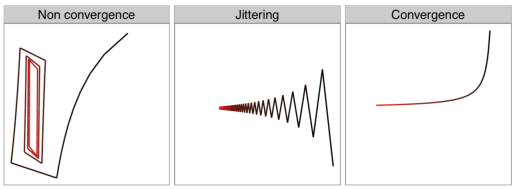

FIGURE 3 – Comportement possibles

est concave et admet comme point fixe 0, de plus

$$\phi \in C([0, r_0[) \cap C^1(]0, r_0[), \phi' > 0$$

On dit que F à la **propriété KL** en $\bar{x}$, $F(\bar{x}) = 0$ s'il existe $\epsilon > 0$ et une fonction dé-singularisante ϕ telle que

$$\|\nabla(\phi \circ F)(x)\| \geq A, \quad \forall x \in B_{\bar{x},\epsilon}, F(\bar{x}) < F(x) < F(\bar{x}) + \epsilon$$

Cette propriété est vraie pour les fonctions semi-algébriques dérivables, pour les fonctions modérées différentiables et les fonctions modérées non-lisses. Si $\nabla F(\bar{x}) \neq 0$, on peut voir ϕ comme une multiplication par une constante positive. Si maintenant F est une fonction analytique

$$F : x \to \sum_{i=l}^{\infty} a_i x^i$$

F est différentiable au voisinage de 0 on aura

$$\phi : t \to \frac{1-\theta}{c} t^{1-\theta}$$

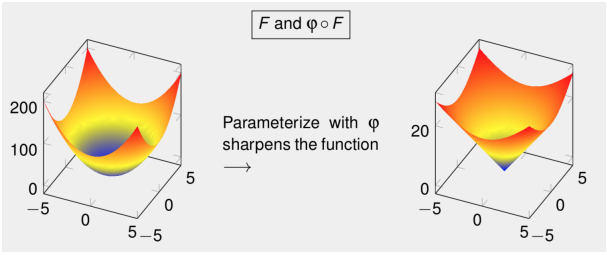

FIGURE 4 – dé-singularisation

Si on ignore la restriction à un voisinage, on peut voir la condition KL comme une généralisation de la convexité forte. Dans le cas d'une forme quadratique

$$F : x \to \frac{1}{2}(x - b^T)A(x - b)$$

avec A symétrique. On peut prendre $\phi(.) = \sqrt{.}$ comme désingularisation. De plus, pour une fonction de Morse C^2 de gradient nul, on a que la Hessienne est non-singulière.

théorème 17. Absil, Mahony, Andrews (2005) *Soit $F : \mathbb{R}^p \to \mathbb{R}$ une fonction minorée, C^1 et de gradient L-Lipschitz vérifiant la propriété KL. On suppose qu'il existe $x_0 \in \mathbb{R}^p$ tel que $\{x \in \mathbb{R}^p | F(x) \leq F(x_0)\}$ est compact et pour tout $k \in \mathbb{N}$*

$$x_{k+1} = x_k - \alpha \nabla F(x_k)$$

avec $\alpha = 1/L$. Alors $x_k \to \bar{x}$ tel que $\nabla F(\bar{x}) = 0$ et $\sum_{k\in\mathbb{N}} \|x_{k+1} - x_k\|$ est fini.

Démonstration. On va utiliser le lemme de descente qui stipule que

$$\forall x, y \quad |F(y) - F(x) - \langle \nabla F(x); y - x\rangle| \leq \frac{L}{2}\|y - x\|^2$$

donc dans ce cas,

$$F(x_{k+1}) \leq F(x_k) - \langle \nabla F(x_k); -\alpha \nabla F(x_k)\rangle + \frac{L\alpha^2}{2}\|\nabla F(x_k)\|^2 = F(x_k) - \frac{1}{2L}\|\nabla F(x_k)\|^2$$

On a donc

$$\sum_{k\in\mathbb{N}} \|\nabla F(x_k)\|^2 \leq F(x_0) - F^*$$

et donc $\|\nabla F(x_k)\| \to 0$. On appelle Ω l'ensemble des points d'accumulations de la suite (x_k). On a immédiatement $\Omega \neq \emptyset$ est compact, F est constante sur Ω et $d(x_k, \Omega) \to 0$. Quelque soit $x \in \Omega$, F satisfait la propriété KL en x. Ainsi il existe $m \in \mathbb{N}$ et $B_i = B_{z_i,\epsilon_i}$ pour $i \in \{1, ..., m\}$ avec $z_i \in \Omega$ un recouvrement de Ω. Pour chaque i, on a ϕ_i tel que

$$\forall x \in B_i, F(z_i) < F(x) < F(z_i) + \epsilon_i, \quad \|\phi_i \circ F(x)\| \geq 1$$

On pose ϵ_0 tel que

$$\epsilon_0 = \inf_{y \in \mathbb{R}^p} d(y, \Omega) > 0$$

ainsi que ϕ tel que

$$\phi = \sum_{i=1}^{m} \phi_i$$

On considère $\epsilon > 0$ tel que

$$\|\nabla \phi \circ F(x)\| \geq 1, \quad \forall x : d(x, \Omega) < \epsilon, F(\bar{x}) < F(x) < F(\bar{x}) + \epsilon$$

Il existe, $k^* \in \mathbb{N}$ tel que $d(x_k, \Omega) < \epsilon$ et $F(\bar{x}) < F(x_k) < F(\bar{x}) + \epsilon$. On obtient alors

$$F(x_{k+1}) \leq F(x_k) - \frac{1}{2}\|x_{k+1} - x_k\|_2 \|\nabla F(x_k)\|$$

et donc, par croissance de ϕ

$$\phi(F(x_{k+1})) \leq \phi(F(x_k) - \frac{1}{2}\|x_{k+1} - x_k\|_2 \|\nabla F(x_k)\|)$$

par concavité

$$\phi(F(x_{k+1})) \leq \phi(F(x_k)) - \phi'(F(x_k))\frac{1}{2}\|x_{k+1} - x_k\|_2 \|\nabla F(x_k)\|)$$

or

$$\phi'(F(x_k))\frac{1}{2}\|x_{k+1} - x_k\|_2 \|\nabla F(x_k)\|) = \frac{1}{2}\|x_{k+1} - x_k\|_2 \|\nabla \phi \circ F(x_k)\|)$$

et donc par propriété KL,

$$\phi(F(x_{k+1})) \leq \phi(F(x_k)) - \frac{1}{2}\|x_{k+1} - x_k\|_2$$

Ce qui donne bien

$$\sum_{k \in \mathbb{N}} \|x_{k+1} - x_k\|_2 \leq 2\phi(F(x_0))$$

On conclut par critère de Cauchy. □

La propriété KL a une généralisation au cas non-lisse.

3.5 Convergence dans le cas bruité : Gradient Stochastique

$$\begin{array}{rccc} F: & \mathbb{R}^p & \to & \mathbb{R} \\ & \theta & \to & \frac{1}{n}\sum_{i=1}^{n} l_i(\theta) \end{array}$$

En supposant que n soit trop grand pour calculer le gradient en entier, on va supposer les (i_k) indépendants et identiquement distribués, on obtient l'algorithme de **descente de gradient stochastique** :

$$\theta_{k+1}|\theta_k = \theta_k - \alpha_k \nabla l_{i_k}(\theta_k)$$

Soit (M_k) une martingale

$$\theta_{k+1}|past = \theta_k - \alpha_k(\nabla F(\theta_k) + M_{k+1})$$

avec $\mathbb{E}[M_{k+1}|past] = 0$. On a plusieurs résultats de la méthode **EDO** qui suppose que le pas est non-sommable mais de carré-sommable. Ljung donne en 1977 que la suite (θ_k) tend vers une limite solution de

$$\theta' = -\nabla F(\theta)$$

On a également un résultat de Benaïm de 1996 reposant sur les notions suivantes. Un ensemble A est dit invariant si pour tout θ_0 dans A il existe une solution à l'EDO précédente telle que $\theta(0) = \theta_0$ et $\theta(\mathbb{R}) \subset A$. A est une chaîne transitive si pour tout $T > 0$ et $\epsilon > 0$ et pour tout $\theta, \mu \in A$ il existe $N \in \mathbb{N}$ et θ_i, pour i allant de 1 à N solutions de l'EDO précédente et $t_i \geq T$ tels que

- $\theta_i(t) \in A$ pour tout $0 \leq t \leq t_i$, $i = 1, ..., N$
- pour tout $i = 1, ..., N-1$, $\|\theta_i(t_i) - \theta_{i+1}(0)\| \leq \epsilon$
- $\|\theta_1(0) - \theta\| \leq \epsilon$ et $\|\theta_N(t_N) - \mu\| \leq \epsilon$

On suppose maintenant également que le moment d'ordre deux du bruit est borné par un certain M. Alors $\sum \alpha_k M_{k+1}$ converge.

théorème 18. Benaïm (1996) *En supposant que les (θ_k) sont bornés, l'ensemble des points d'accumulations est compact, connexe et est une chaîne transitive invariante presque sûrement.*

Proposition 19. *En particulier, si F est une fonction modérée, alors tout compact, chaîne transitive A alors*

- *F est constante sur A*
- *pour tout $\theta \in A$, on a $\nabla F(\theta) = 0$*

Démonstration. Pour prouver ce résultat on va montrer que $L^- = \min_{\theta \in A} F(\theta)$ et $L^+ = \max_{\theta \in A} F(\theta)$ sont égaux. On pose $L^- = 0$ et on rappelle que γ-Lipschitz on A. On rappelle alors le théorème de Morse-Sard.

théorème 20. Morse-Sard : *Pour $\delta > 0$ tel que $]0; 2\delta]$ ne contient aucun point critique de F. Alors, pour $\epsilon = \delta/\gamma$, et $x, y \in A$ tels que $\|x - y\| \leq \epsilon$ implique $|f(x) - f(y)| \leq \delta$. On pose*

$$\Delta = \min_{\theta \in A, \delta \leq F \leq 2\delta} \|\nabla F(\theta)\|_2 > 0$$

et $T = \delta/\Delta^2$.

Pour chaque $\theta_0 \in A$ tel que $F(\theta_0) \leq 2\delta$ toute solution de $\dot{\theta} = \nabla F(\theta)$ partant de θ_0 vérifie $F(\theta(t)) \leq \delta$ pour tout $t \geq T$.
Pour tout $\theta, \mu \in A$ avec $F(\theta) = 0$, par transitivité $F(\mu) \leq 2\delta$. En faisant tendre δ vers 0 on a alors $F(\mu) = 0$ pour tout $\mu \in A$ et donc $L^+ = 0$. □

3.6 Extension au cas non-lisse

Maintenant on écrit

$$\theta_{k+1}|past = \theta_k - \alpha_k(v + M_{k+1})$$

avec $v \in \partial F(\theta_k)$ où ∂ est une généralisation du gradient. C'est la base de la méthode de **descente de sous-gradient stochastique**. Il existe plusieurs choix possibles pour ∂

- Convexe : $\partial_{conv}F(x) = \{v \in \mathbb{R}^p | F(y) \geq F(x) + v^T(y-x), \forall y \in \mathbb{R}^p\}$
- Frechet : $\partial_{fre}F(x) = \{v \in \mathbb{R}^p | \lim_{y \to x} \inf_{y \neq x} \frac{F(y)-F(x)-v^T(y-x)}{\|y-x\|}\}$
- Limite : $\partial_{lim}F(x) = \{v \in \mathbb{R}^p | \exists (y_k, v_k), y_k \to x, v_k \to v, v_k \in \partial_{fre}F(y_k)\}$
- Clarke : $\partial_{cla}F(x) = conv(\partial_{lim}F(x))$

Prenons par exemple la fonction valeur absolue dans $\mathbb{R}$ alors

$$\begin{array}{rcl} \partial_{conv}F(x) & = & \emptyset \\ \partial_{fre}F(x) & = & \emptyset \\ \partial_{lim}F(x) & = & \{-1, 1\} \\ \partial_{cla}F(x) & = & [-1; 1] \end{array}$$

4 Paysage de la fonction objectif

On rappelle la décomposition du risque dans la figure 5. Et nous allons nous concentrer sur l'erreur d'approximation. L'erreur d'approximation est définie par $R(\tilde{f}_{w^*})) - R^*$ et

$0 \leq$	$R(\tilde{f}_{\mathbf{w}})$	$-\ R^*$	$=$	(excès de risque)
	$R(\tilde{f}_{\mathbf{w}^*})$	$-\ R^*$	$+$	(erreur d'approximation)
	$\hat{R}(\tilde{f}_{\mathbf{w}^*})$	$-\ R(\tilde{f}_{\mathbf{w}^*})$	$+$	(erreur de généralisation)
	$\hat{R}(\tilde{f}_{\mathbf{w}'})$	$-\ \hat{R}(\tilde{f}_{\mathbf{w}^*})$	$+$	$\leq \varepsilon$
	$\hat{R}(\tilde{f}_{\mathbf{w}})$	$-\ \hat{R}(\tilde{f}_{\mathbf{w}'})$	$+$	(erreur d'optimisation)
	$R(\tilde{f}_{\mathbf{w}})$	$-\ \hat{R}(\tilde{f}_{\mathbf{w}})$		(erreur de généralisation)

FIGURE 5 – Décomposition du risque

on a la propriété suivante $R(\tilde{f}_{w^*})) \leq \inf_w R(\tilde{f}_w)) + \epsilon$. On veut calculer un certain w^{calc} qui minimise :

$$\hat{R}(\tilde{f}_w^{\text{calc}})) - \min_w \hat{R}(\tilde{f}_w))$$

Idéalement cette grandeur est nulle ou au moins majorée. On suppose que la fonction $w \to \hat{R}(\tilde{f}_w))$ est C^2 (ce qui n'est pas le cas en général en pratique). On a alors le développement de Taylor suivant

$$\hat{R}(\tilde{f}_w)) = \hat{R}(\tilde{f}_{w^*})) + V_w\hat{R}(\tilde{f}_{w^*}))(w - w^*) + \frac{1}{2}V_w^2\hat{R}(\tilde{f}_{w^*}))(w - w^*)^2 + o(\|w - w^*\|^2)$$

On distingue alors

— **minimiseur global** - $\forall w, \hat{R}(\tilde{f}_{w^*})) \leq \hat{R}(\tilde{f}_w))$
— **minimiseur local** - pour un certain ouvert $\mathcal{O}$ on a $\forall w \in \mathcal{O}, \hat{R}(\tilde{f}_{w^*})) \leq \hat{R}(\tilde{f}_w))$
— **point critique du second ordre** - $V\hat{R}(\tilde{f}_{w^*})) = 0$ et $V^2\hat{R}(\tilde{f}_{w^*})) \geq 0$
— **point critique du premier ordre** - $V\hat{R}(\tilde{f}_{w^*})) = 0$

Les limites des algorithmes d'optimisation sont souvent des points critiques du second ordre. On parle parfois de plateau, ou encore de bad saddle point (mauvais point selle). De plus, un point critique du premier ordre qui n'est pas du second ordre est parfois appelé **strict saddle** (strictement selle). On a les relations suivantes

$$[\text{min global}] \Rightarrow [\text{min local}] \Rightarrow [\text{PC du } 2^{nd} \text{ ordre}] \Rightarrow [\text{PC du } 1^{er} \text{ ordre}]$$

Sans hypothèse de convexité, on n'a pas de réciprocité dans les relations précédentes. On a vu dans le chapitre précédent, que dans un cadre assez vaste l'algorithme du gradient converge vers un point critique du premier ordre. Dans un cadre plus restreint, l'algorithme du gradient converge vers un point critique du second ordre. Pour le gradient stochastique, que l'on modélise comme suit

$$w^{k+1} = w^k - s(\nabla E(w^k) + b_k)$$

avec b_k un bruit (que l'on avait noté M_{k+1} dans la section précédente), on peut montrer que grâce au bruit on peut sortir des plateaux par marche aléatoire et donc converger vers un minimum local. Comme on ne connaît pas le nombre de plateaux, on ne peut pas donner de résultats de temps de convergence. **Pour en savoir plus voir : S. Arora** qui a été conférencier à ICM (conférence internationale sur les mathématiques).

4.1 Paysage pour les réseaux larges

On considère un problème de régression, avec un réseau feed forward, fully connected, et des observations $(x_l, y_l)_{l \in [\![1;L]\!]}$.

théorème 21. Nguyen, Hein 2017 ICML (simplifié) : *On suppose $\sigma \in C^1$, que pour tout $t \in \mathbb{R}$ on a $\sigma'(t) \neq 0$. On suppose que le coût l est C^1 à valeurs dans $\mathbb{R}_+$ et est nul en 0. On suppose également que $\nabla l(y) = 0$ si et seulement si $y = 0$. On note $X = (x_1, ..., x_L) \in \mathbb{R}^{n_0 \times L}$ et $A = \begin{pmatrix} X \\ 1_L^T \end{pmatrix}$. On considère un point critique du premier ordre (w, b) de $\hat{R}$. On suppose que $rg(A) = L$ et que pour tout $h \in \{1, ..., H\}$ on a $rg(W_h(w_h)) = n_h$. Alors on a*

$$\hat{R}(w, b) = 0$$

et (w, b) est un minimum global.

Les hypothèses contraignantes sont celles sur le rang. Elles impliquent en autre

$$L \leq n_0 + 1 \quad \text{et} \quad n_H \leq n_{H-1} \leq ... \leq n_0$$

Démonstration. On rappelle que pour un réseau feed forward, fully connected, pour $h \in [\![1; H]\!]$, $i \in [\![1; n_h]\!]$, $j \in [\![1; n_{h-1}]\!]$ et pour $k = (i-1)n_{h-1} + j$ on a

$$\frac{\partial E}{\partial w_{h,k}}(w, b) = \sum_{l=1}^{L} \left[f_{w,b}^{h-1}(x_l)\right]_j \left[d_{w,b}^h f_{w,b}^{h-1}(x_l)\right]_i \left[\Delta^h(x_l)\right]_i$$

et

$$\frac{\partial E}{\partial b_{h,i}}(w, b) = \sum_{l=1}^{L} [d_{w,b}^h(f_{w,b}^{h-1}(x))]_i \Delta_i^h(x_l)$$

avec

$$\Delta^h(x_l) = \begin{cases} \nabla l(f_{w,b}(x_l) - y_l) & \text{si } h = H \\ [W_{h+1}(w_{h+1})]^T.\text{diag}(d_{w,b}^{h+1}(f_{w,b}^h(x_l))).\Delta^{h+1} & \text{sinon} \end{cases}$$

On note

$$\overline{\Delta}_i^k(x_l) = \left[d_{w,b}^h f_{w,b}^{h-1}(x_l)\right]_i \Delta_i^h(x_l)$$

On va montrer par récurrence que pour tout $h \in [\![1; H]\!]$ et tout $l \in\in [\![1; L]\!]$ on a $\overline{\Delta}^k(x_l) = 0$. En effet, pour $h = 1$ on a

$$0 = \frac{\partial E}{\partial w_{h,k}}(w, b) = \sum_{l=1}^{L} \left[f_{w,b}^0(x_l)\right]_j \overline{\Delta}_i^1(x_l)$$

et

$$0 = \frac{\partial E}{\partial b_{h,i}}(w, b) = \sum_{l=1}^{L} \overline{\Delta}_i^1(x_l)$$

Ces équations sont linéaires et on les réécrit sous forme matricielle en posant

$$\overline{\Delta}^1 = \begin{pmatrix} \overline{\Delta}_1^1(x_1) & \dots & \overline{\Delta}_{n_1}^1(x_1) \\ \vdots & & \vdots \\ \overline{\Delta}_1^1(x_L) & \dots & \overline{\Delta}_{n_1}^1(x_L) \end{pmatrix}$$

et on obtient

$$0 = A\overline{\Delta}^1$$

Or, on a supposé $rg(A) = L$ et donc $A^T A$ est inversible et ainsi $\overline{\Delta}^1 = 0$. Supposons maintenant, la propriété vraie pour un certain $h \in [\![1; H]\!]$. Comme $\sigma'(t) \neq 0$ pour tout $t \in \mathbb{R}$, $\text{diag}(d_{w,b}^{h+1}(f_{w,b}^h(x_l)))$ est inversible. Donc, par rétro-propagation

$$0 = \Delta^h(x_l) = [W_{h+1}(w_{h+1})]^T \overline{\Delta}^{h+1}(x_l)$$

et comme $W_{h+1}(w_{h+1}$ est de rang n_{h+1} on a bien $\overline{\Delta}^{h+1} = 0$. Ce qui achève la récurrence. On en déduit que $\overline{\Delta}^H(x_l) = 0$ quelque soit $l \in [\![1; L]\!]$, à nouveau comme $\sigma'(t) \neq 0$ pour tout $t \in \mathbb{R}$ on a $\Delta^H(x_l) = 0$, i.e.

$$\nabla(l(f_{w,b}(x_l) - y_l) = 0$$

On obtient donc $l(f_{w,b}(x_l) - y_l = 0$ et

$$\hat{R}(w, b) = 0$$

□

Les points clefs de la preuves sont au nombre de deux. On n'utilise ici pas grand chose de plus que les formules de rétro-propagation. Cependant, numériquement avoir la condition sur σ' est rarement vérifiée. Le message principal derrière cette preuve : en cas d'échec d'optimisation il faut augmenter la largeur du réseau. Notons qu'ici on peut supposer ici que n_0 correspond non-pas à la taille des données mais la taille de l'embedding de données via des couches antérieurs du réseau ou d'une autre méthode d'embedding. Ce théorème donne de plus une borne sur la taille des couches. En pratique, une autre méthode consiste à augmenter la profondeur du réseau. La régularisation est cachée dans l'hypothèse $rg(W_h(w_h)) = n_h$.

4.2 Paysage pour les réseaux linéaires

On considère un problème de régression, avec un réseau feed forward, fully connected, linéaire et sans biais, ainsi que des observations $(x_l, y_l)_{l\in[\![1;L]\!]}$. On simplifie le problème en posant $H = 2$ (il y a possibilité de généraliser à $H \geq 2$). L'intérêt de ces simplifications est de pouvoir tout caractériser dans le paysage de la fonction objectif. On note $A \in \mathbb{R}^{n_2\times n_1}$ et $B \in \mathbb{R}^{n_1\times n_0}$, i.e.

$$E(A, B) = \sum_{l=1}^{L} \|y_l - ABx_l\|^2$$

et on note le problème intermédiaire F,

$$F(D) = \sum_{l=1}^{L} \|y_l - Dx_l\|^2$$

avec $D \in \mathbb{R}^{n_2\times n_0}$ et on note enfin

$$\begin{array}{lll} \Sigma_{XX} = \sum_{l=1}^{L} x_l x_l^T \in \mathbb{R}^{n_0\times n_0} & , & \Sigma_{XY} = \sum_{l=1}^{L} x_l y_l^T \in \mathbb{R}^{n_0\times n_2} \\ \Sigma_{YX} = \sum_{l=1}^{L} y_l x_l^T \in \mathbb{R}^{n_2\times n_0} & , & \Sigma_{YY} = \sum_{l=1}^{L} y_l y_l^T \in \mathbb{R}^{n_2\times n_2} \end{array}$$

On a quelques remarques intéressantes qui formes trois propriétés préliminaires.

- Pour toute matrice $C \in \mathbb{R}^{n_1\times n_1}$ inversible, on a $AB = (AC)(C^{-1}B) = A'B'$.
- Si Σ_{XX} est inversible, alors $\Sigma_{YX}\Sigma_{XX}^{-1} \in \arg\min_D F(D)$.
- Soit $M \in \mathbb{R}^{n\times p}$ avec $p \leq n$ et rang p. Pour tout $x \in \mathbb{R}^n$ la projection $P_M(x)$ de x sur l'espace vectoriel généré par les colonnes de M vaut $P_M(x) = M(M^TM)^{-1}M^Tx$.

Démonstration. Montrons le dernier point. On a par définition de la projection

$$P_M(x) = \arg\min_{z=My} \|x - My\|^2$$

i.e.

$$P_M(x) = M \arg\min_y \|x - My\|^2 = M \arg\min_y f(y)$$

or f est quadratique et

$$\nabla f(y) = 2M^T(My - x) \qquad \text{et} \qquad \nabla^2 f(y) = 2M^T M$$

La hessienne de f est défini positive. L'unique minimiseur de f correspond à son unique point critique et vérifie

$$M^T My = M^T x$$

on a donc

$$P_M(x) = M(M^T M)^{-1} M^T x$$

□

On va se donner deux lemmes pour approfondir ces remarques. On va toujours faire l'hypothèse que A est de rang n_1. En effet, les matrices ne vérifiant pas cela est un ensemble de mesure nulle. De plus, cet ensemble ne contient au mieux que des points critiques du second ordre.

Lemma 22. *Si Σ_{XX} est inversible. Soient $A \in \mathbb{R}^{n_2 \times n_1}$ de rang n_1 et $B \in \mathbb{R}^{n_1 \times n_0}$. Alors (A, B) est un point critique du premier ordre de E si et seulement si*

$$AB\Sigma_{XX}B^T = \Sigma_{YX}B^T \qquad et \qquad B = (A^T A)^{-1} A^T \Sigma_{YX} \Sigma_{XX}^{-1}$$

Démonstration. On fait le développement limité de E sur sa seconde variable

$$E(A, B + B') = \sum_{l=1}^{L} \|y_l - A(B + B')x_l\|^2$$

i.e.

$$E(A, B + B') = E(A, B) - 2\sum_{l=1}^{L} \langle y_l - ABx_l; AB'x_l \rangle + o(\|B'\|)$$

$$E(A, B) - 2\langle \sum_{l=1}^{L} A^T ABx_l x_l^T - \sum_{l=1}^{L} A^T y_l x_l^T; B' \rangle + o(\|B'\|)$$

On en déduit

$$\frac{\partial E}{\partial B}(A, B) = 2(A^T AB\Sigma_{XX} - A^T \Sigma_{YX}$$

de même

$$\frac{\partial E}{\partial A}(A, B) = 2(AB\Sigma_{XX}B^T - \Sigma_{YX}B^T$$

Et donc pour (A, B) point critique, on trouve bien

$$AB\Sigma_{XX}B^T = \Sigma_{YX}B^T \qquad \text{et} \qquad B = (A^T A)^{-1} A^T \Sigma_{YX} \Sigma_{XX}^{-1}$$

□

Le lemme suivant est le pivot de la preuve de ce qui suit en donnant un résultat de commutativité. En effet, deux matrices commutent si elles sont co-diagonalisable.

Lemma 23. *Si Σ_{XX} est inversible. Soient $A \in \mathbb{R}^{n_2 \times n_1}$ de rang n_1 et $B \in \mathbb{R}^{n_1 \times n_0}$. Alors les deux assertions suivantes sont équivalentes*

- *(A, B) est un point critique du premier ordre de E.*
- *$AB = P_A \Sigma_{YX} \Sigma_{XX}^{-1}$ et pour $\Sigma = \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}$ on a $P_A\Sigma = P_A\Sigma P_A = \Sigma P_A$ (**commutativité**).*

Démonstration. On commence par montrer $1 \Rightarrow 2$. Soit (A, B) tel que

$$AB\Sigma_{XX}B^T = \Sigma_{YX}B^T \qquad \text{et} \qquad B = (A^TA)^{-1}A^T\Sigma_{YX}\Sigma_{XX}^{-1}$$

On a donc, par la propriété préliminaire 3

$$AB = P_A\Sigma_{YX}\Sigma_{XX}^{-1}$$

De plus, on a

$$AB\Sigma_{XX}B^TA^T = \Sigma_{XX}B^TA^T$$

i.e.

$$P_A\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XX}P_A = \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XX}P_A$$

Ainsi,

$$P_A\Sigma P_A = \Sigma P_A$$

comme Σ et P_A sont symétriques, on a aussi

$$P_A\Sigma = P_A^T\Sigma^T = (\Sigma P_A)^T = (P_A\Sigma P_A)^T = P_A^T\Sigma^T P_A^T = P_A\Sigma P_A = \Sigma P_A$$

Il reste à montrer la réciproque $1 \Leftarrow 2$. On suppose que $AB = P_A\Sigma_{YX}\Sigma_{XX}^{-1}$ et $P_A\Sigma = P_A\Sigma P_A = \Sigma P_A$. En multipliant par à gauche la $(A^TA)^{-1}A^T$, on obtient

$$(A^TA)^{-1}A^TAB = (A^TA)^{-1}A^TA(A^TA)^{-1}A^T\Sigma_{YX}\Sigma_{XX}^{-1}$$

i.e.

$$B = (A^TA)^{-1}A^T\Sigma_{YX}\Sigma_{XX}^{-1}$$

On sait que $P_A\Sigma P_A = \Sigma P_A$ i.e.

$$P_A\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XX}P_A = \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XX}P_A$$

puis comme $AB = P_A\Sigma_{YX}\Sigma_{XX}^{-1}$, on obtient

$$AB\Sigma_{XX}B^TA^T = \Sigma_{XX}B^TA^T$$

En multipliant à droite par $A(A^TA)^{-1}$, on trouve

$$AB\Sigma_{XX}B^TA^TA(A^TA)^{-1} = \Sigma_{XX}B^TA^TA(A^TA)^{-1}$$

i.e.

$$AB\Sigma_{XX}B^T = \Sigma_{XX}B^T$$

et donc on retrouve le résultat du lemme précédent. □

On diagonalise Σ qui est symétrique réelle.

$$\Sigma = U\Lambda U^T$$

avec $\Lambda = \mathbb{R}^{n_2 \times n_2}$ diagonale, et $U = \mathbb{R}^{n_2 \times n_2}$ unitaire. Pour $\mathcal{S} \subset [\![1; n_2]\!]$, on note $U_\mathcal{S}$ la matrice extraite de U en prenant les colonnes d'indice dans $\mathcal{S}$.

Proposition 24. *Si Σ_{XX} est inversible. Soient $A \in \mathbb{R}^{n_2 \times n_1}$ de rang n_1 et $B \in \mathbb{R}^{n_1 \times n_0}$. On suppose que les valeurs propres de Σ sont distinctes. Alors (A, B) est un point critique du premier ordre de E si et seulement s'il existe $C \in \mathbb{R}^{n_1 \times n_1}$ inversible et $\mathcal{S} \subset [\![1; n_2]\!]$ de taille n_1 tels que*

$$A = U_\mathcal{S} C \qquad et \qquad B = C^{-1} U_\mathcal{S}^T \Sigma_{YX} \Sigma_{XX}^{-1}$$

Démonstration. On commence par montrer le sens réciproque ($\Leftarrow$). On a

$$U_\mathcal{S}^T U_\mathcal{S} = \mathrm{Id}_{n_1}$$

Donc $P_{U_\mathcal{S}} = U_\mathcal{S} U_\mathcal{S}^T$. Si $A = U_\mathcal{S} C$ et $B = C^{-1} U_\mathcal{S}^T \Sigma_{YX} \Sigma_{XX}^{-1}$ on a

$$AB = U_\mathcal{S} C C^{-1} U_\mathcal{S}^T \Sigma_{YX} \Sigma_{XX}^{-1} = P_{U_\mathcal{S}} \Sigma_{YX} \Sigma_{XX}^{-1}$$

Or C est inversible et $A = U_\mathcal{S} C$ donc

$$AB = P_A \Sigma_{YX} \Sigma_{XX}^{-1}$$

Il faut encore montrer que $P_A \Sigma = P_A \Sigma P_A = \Sigma P_A$. Comme $P_{U_\mathcal{S}} = U_\mathcal{S} U_\mathcal{S}^T$ a n_1 valeurs propres valant 1, auxquelles on peut associer comme vecteurs propres les colonnes de $U_\mathcal{S}$. Elle a $n_2 - n_1$ valeurs propres égales à 0 auxquelles on peut associer les autres colonnes de U, i.e.

$$P_A = P_{U_\mathcal{S}} = U \mathrm{diag}(1_\mathcal{S}) U^T$$

on a alors

$$P_A \Sigma = P_A \Sigma P_A = \Sigma P_A$$

On en conclut que (A, B) est un point critique du premier ordre de E. Il reste à montrer le sens direct ($\Rightarrow$). On a alors $P_A \Sigma = \Sigma P_A$ et P_A, Σ sont diagonalisables et donc co-diagonalisables. Ainsi il existe V unitaire telle que

$$P_A = V \Lambda_{P_A} V^T \quad \text{et} \quad \Sigma = V \Lambda_\Sigma V^T$$

Puisque les valeurs propres de Σ sont distinctes, quitte à changer l'ordre des colonnes de V et leur signe on obtient $U = V$ et ainsi

$$P_A = U \Lambda_{P_A} U^T$$

Or P_A est une projection, il existe donc $\mathcal{S} \subset [\![1; n_2]\!]$ tel que

$$P_A = U \mathrm{diag}(1_\mathcal{S}) U^T = P_{U_\mathcal{S}}$$

Il existe donc une matrice inversible C telle que

$$A = U_{\mathcal{S}} C$$

et

$$B = (A^T A)^{-1} A^T \Sigma_{YX} \Sigma_{XX}^{-1} = C^{-1} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1}$$

□

On suppose que les valeurs propres sont ordonnées, i.e.

$$\lambda_1 > \lambda_2 > ... > \lambda_{n_2}$$

Proposition 25. *Si Σ_{XX} est inversible. Soient $A \in \mathbb{R}^{n_2 \times n_1}$ de rang n_1 et $B \in \mathbb{R}^{n_1 \times n_0}$. On suppose que les valeurs propres de Σ sont distinctes. Alors,*

— *pour tout point critique du premier ordre (A, B) de E et pour $\mathcal{S}$ définissant A et B comme dans la proposition précédente. On a*

$$AB = P_{U_{\mathcal{S}}} \Sigma_{YX} \Sigma_{XX}^{-1} \text{ et } E(A, B) = tr(\Sigma_{YY}) - \sum_{i \in \mathcal{S}} \lambda_i$$

— *(A, B) est un minimiseur global si et seulement si c'est un point critique du premier ordre associé à $\mathcal{S} = [\![1; n_1]\!]$.*

— *(A, B) est un minimiseur local si et seulement si c'est un minimiseur global.*

Démonstration. On va d'abord montrer le premier point. Avec les résultats précédents, on a

$$A = U_{\mathcal{S}} C \quad \text{et} \quad B = C^{-1} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1}$$

et donc

$$AB = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1}$$

et enfin $U_{\mathcal{S}} U_{\mathcal{S}}^T = P_{\mathcal{S}}$. Avant de montrer le second résultat du premier point, rappelons le préliminaire suivant

$$u^T v = \text{tr}(vu^T) = \text{tr}(uv^T) = v^T u$$

On a

$$E(A, B) = \sum_{i=1}^{L} \|y_i - ABx_i\|^2 = \sum_{i=1}^{L} (y_i - ABx_i)^T (y_i - ABx_i)$$

$$= \sum_{i=1}^{L} y_i^T y_i - 2x_i^T B^T A^T y_i + x_i^T B^T A^T ABx_i$$

$$= \sum_{i=1}^{L} y_i^T y_i - 2\text{tr}(B^T A^T \Sigma_{YX}) + \text{tr}(B^T A^T AB \Sigma_{XX})$$

On va traiter séparément les deux traces.

$$\mathrm{tr}(B^T A^T \Sigma_{YX}) = \mathrm{tr}(\Sigma_{XX}^{-1}\Sigma_{XY}P_{U_\mathcal{S}}\Sigma_{YX}) = \mathrm{tr}(\Sigma P_{U_\mathcal{S}}) = \sum_{i\in\mathcal{S}}\lambda_i$$

Enfin,

$$B^T A^T AB\Sigma_{XX} = B^T A^T P_{U_\mathcal{S}}\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XX} = \Sigma_{XX}^{-1}\Sigma_{XY}U_\mathcal{S}U_\mathcal{S}^T P_{U_\mathcal{S}}\Sigma_{YX}$$

$$= \Sigma_{XX}^{-1}\Sigma_{XY}U_\mathcal{S}U_\mathcal{S}^T U_\mathcal{S}U_\mathcal{S}^T P_{U_\mathcal{S}}\Sigma_{YX} = \Sigma_{XX}^{-1}\Sigma_{XY}U_\mathcal{S}U_\mathcal{S}^T P_{U_\mathcal{S}}\Sigma_{YX} = B^T A^T \Sigma_{YX}$$

On en déduit donc

$$E(A,B) = \mathrm{tr}(\Sigma_{YY}) - \sum_{i\in\mathcal{S}}\lambda_i$$

Le second point est une conséquence immédiate du premier point et des propositions précédentes. On va montrer qu'un point critique qui n'est pas un minimum global n'est pas un minimum local. Soit (A,B) un point critique associé à $\mathcal{S}$ vérifiant

$$\exists i \in [\![1;n_1]\!] \text{ et } i\notin\mathcal{S}, \qquad \exists i \in [\![n_1+1;n_2]\!] \text{ et } j\in\mathcal{S}$$

et tel qu'il existe une matrice inversible C telle que

$$A = U_\mathcal{S}C \quad \text{et} \quad B = C^{-1}U_\mathcal{S}^T\Sigma_{YX}\Sigma_{XX}^{-1}$$

On note u_x la $x^{\text{ème}}$ colonne de U et

$$u_t = \frac{u_j + tu_i}{\sqrt{1+t^2}}$$

On considère $U_t = U_\mathcal{S}$ sauf la colonne j qui vaut u_t et on pose

$$A_t = U_tC \quad \text{et} \quad B = C^{-1}U_t^T\Sigma_{YX}\Sigma_{XX}^{-1}$$

On a $U_t^TU_t = \mathrm{Id}_{n_1}$ et donc $U_tU_t^T = P_{U_t}$ et on trouve

$$P_{U_t}U = U\Sigma'$$

avec

$$\Sigma'_{k,l} = \begin{cases} 1 & \text{si } k=l\in\mathcal{S}\backslash\{j\} \\ \frac{t}{1+t^2} & \text{si } l=j, k=i \\ \frac{1}{1+t^2} & \text{si } l=j, k=j \\ \frac{t^2}{1+t^2} & \text{si } l=i, k=i \\ \frac{t}{1+t^2} & \text{si } l=i, k=j \\ 0 & \text{sinon} \end{cases}$$

Comme précédemment on obtient

$$A_tB_t = P_{U_t}\Sigma_{YX}\Sigma_{XX}^{-1}$$

ainsi que,

$$E(A_t, B_t) = \mathrm{tr}(\Sigma_{YY}) - 2\mathrm{tr}(B_t^T A_t^T \Sigma_{YX}) + \mathrm{tr}(B_t^T A_t^T A_t B_t \Sigma_{XX})$$

et

$$B_t^T A_t^T A_t B_t \Sigma_{XX} = B_t^T A_t^T \Sigma_{YX}$$

Calculons $\mathrm{tr}(B_t^T A_t^T \Sigma_{YX})$

$$\mathrm{tr}(B_t^T A_t^T \Sigma_{YX}) = \mathrm{tr}(\Sigma P_{U_t}) = \sum_{k \in \mathcal{S}\setminus\{j\}} \lambda_k + \frac{t^2}{1+t^2}\lambda_i + \frac{1}{1+t^2}\lambda_j = \sum_{k\in\mathcal{S}} \lambda_k + t^2(\lambda_i - \lambda_j) + o(t^2)$$

On en déduit

$$E(A_t, B_t) = E(A, B) - t^2(\lambda_i - \lambda_j) + o(t^2)$$

et comme $\lambda_i > \lambda_j$ on conclut que (A, B) n'est pas un minimum local. □

5 Compromis approximation-estimation-optimisation

On considère un échantillon $(X_i, Y_i)_{i\in[\![1;n]\!]} \sim P_{X,Y}$ i.i.d. et une fonction mesurable g. On rappelle la définition des notions de risque et de risque empirique de g relativement à une perte positive L

$$R(g) = \mathbb{E}[L(g(X), Y)] \qquad \text{et} \qquad \hat{R}(g) = \frac{1}{n}\sum_{i=1}^{n} L(g(X_i), Y_i)$$

Le risque de de Bayes $R^* = \inf_g R(g)$ sur l'ensemble des fonctions mesurables. Les paramètres quasi-optimaux w^* du réseau pour le risque R et $\hat{w}$ pour le risque empirique. On a

$$R(f_{w^*}) \leq \inf_w R(f_w) + \epsilon \qquad \text{et} \qquad \hat{R}(f_{\hat{w}}) \leq \inf_w \hat{R}(f_w) + \epsilon$$

On s'intéresse à l'erreur d'approximation définie par $R(f_{w^*}) - R^*$ comme dans la figure 1. Cette erreur soulève la question suivante : quelles fonctions peuvent être approchées par des réseaux de neurones ? On parle d'**expressivité**.

5.1 Expressivité des réseaux de neurones

Nous allons étudier les capacités d'approximation des réseaux de neurones feedforward. On a plusieurs questions naturelles

- Quelles fonctions peut-on approcher avec un réseau à k couches cachées ?
- Une seule couche cachée est-elle suffisante ?
- L'expressivité augmente-t-elle avec la profondeur
- Pour un nombre de neurones donné, vaut-il mieux un réseau peu profond et large ou un réseau profond et étroit ?

Nous allons répondre partiellement à ces questions. Commençons par un cas simple, soient $f : [a;b]^d \to \mathbb{R}$ continue et $\epsilon > 0$. Par le théorème de Heine, on a l'uniforme continuité de f et donc il existe $\delta > 0$ tel que

$$\|x - y\|_\infty \leq \delta \Rightarrow |f(x) - f(y)| \leq \epsilon$$

On découpe $[a;b]^d$ en N^d cubes A_i de largeur $(b-a)/N \simeq \delta$, avec N entier. Sur chaque cube A_i, on approche f par la valeur $f(c_i)$ en son centre c_i. On a alors

$$\sup_{x\in[a;b]^d} \left| f(x) - \sum_{i=1}^{N^d} f(c_i)\mathbf{1}_{A_i} \right|$$

ainsi on a le résultat

théorème 26. *Toute fonction réelle continue sur $[a;b]^d$ peut-être arbitrairement bien approchée (au sens de la norme infinie) par une fonction constante par morceaux.*

Soit, maintenant, $f : [0;2\pi] \to \mathbb{R}$ une fonction de carré intégrable. Considérons ses coefficients de Fourier

$$\hat{f}(n) = \frac{1}{2\pi} \int_0^{2\pi} f(x)e^{-ixn}dx$$

et les sommes partielles $S_N : [0;2\pi] \to \mathbb{R}$ de la série de Fourier

$$S_N(x) = \sum_{n=-N}^{N} \hat{f}(n)e^{inx}$$

D'après le théorème de Riesz-Fischer, $S_N \to f$ dans $L^2([0;2\pi],\mathbb{R})$ quand N tend vers $+\infty$.

théorème 27. *L'ensemble des fonctions de la forme*

$$x \to \sum_{n=-N}^{N} c_n e^{inx}$$

est dense $L^2([0;2\pi],\mathbb{R})$.

Les deux résultats précédents sont assez simples et connus, ils donnent un petit échauffement. On va voir un exemple de représentation (et non d'approximation) exacte avec un nombre fini de termes.

théorème 28. Kolmogorov-Arnold (1956) : *Toute fonction continue $f : [0;2\pi] \to \mathbb{R}$ de plusieurs variables peut être représentée comme une superposition finie de fonctions continues d'une variable $\Phi_i, \Psi_i : \mathbb{R} \to \mathbb{R}$ et de l'opération somme :*

$$f(x_1, ..., x_d) = \sum_{i=0}^{2d} \Phi_i \left(\sum_{j=1}^{d} \Psi_{i,j}(x_j) \right)$$

Notons que ça ressemble un peu à un réseau feedforward à 2 couches cachées. Mais, pour les réseaux feedforward, on impose une unique fonction d'activation et donc on se contentera d'une représentation approximative. On peut voir un lien avec le $13^{\text{ième}}$ problème de Hilbert. Il est formulé dans une liste de 23 problèmes en 1900 par David Hilbert, cherche à savoir si on peut exprimer une solution $x(a,b,c)$ de l'équation

$$x^7 + ax^3 + bx^2 + cx + 1 = 0$$

à l'aide d'une superposition finie de fonctions algébriques/continues de deux variables. Il était conjecturé que cela n'était pas possible. Kolmogorov et Arnold ont contredit cette conjecture en 1956 en montrant qu'en fait, toute fonction continue de plusieurs variables pouvait s'exprimer comme une superposition d'un nombre fini de fonctions continues de 2 variables.

5.2 Le théorème d'approximation de Cybenko

On s'intéresse aux réseaux de neurones feedforward à 1 couche cachée, i.e., aux fonctions $f : \mathbb{R}^d \to \mathbb{R}$ de la forme

$$f(x) = \sum_{i=1}^{N} v_i \sigma(\langle w_i; x\rangle + b_i)$$

avec $\sigma : \mathbb{R} \to \mathbb{R}$ une fonction d'activation. Pour la preuve nous aurons besoin des théorèmes suivant

théorème 29. Hahn-Banach (propriété de séparation) : *Soient X un $\mathbb{K}$-espace vectoriel ($\mathbb{K} = \mathbb{R}$ ou $\mathbb{C}$) topologique (l'addition et la multiplication par un scalaire sont continues) localement convexe, M un sous-espace de X et $x_0 \in X \backslash \overline{M}$. Alors il existe une forme linéaire continue $L : X \to \mathbb{K}$ telle que*

$$L(x_0) = 1 \qquad et \qquad M \subset L^{-1}(\{0\})$$

Soit X espace topologique séparé et localement compact et $f : X \to \mathbb{C}$ on dit que f s'annule à l'infini si et seulement si pour tout $\epsilon > 0$ il existe un compact $K \subset X$ tel que $|f(x)| \leq \epsilon$ pour tout $x \in X \backslash K$. On note $C_0(X)$ l'ensemble des fonctions continues de X dans $\mathbb{C}$ qui s'annulent à l'infini. Notons que si X est séparé et compact, alors $C_0(X) = C(X)$. Soit $(X, \mathcal{M})$ un espace mesurable, une mesure est dite **complexe** μ sur $\mathcal{M}$ satisfait

— pour tout $E \in \mathbb{M}$ et toute partition mesurable $(E_i)_{i\in\mathbb{N}}$ de E, $\mu(E) = \sum_{i\in\mathbb{N}} \mu(E_i)$. (en particulier on peut renuméroter les termes de la série et donc elle est absolument convergente)

Une mesure signée correspond au cas non-complexe. On associe à μ sa **mesure de variation totale**, notée $|\mu|$, définie par

$$\forall E \in \mathcal{M} |\mu|(E) = \sup_{E=\cup_{i\in\mathbb{N}} E_i} \sum_{i\in\mathbb{N}} |\mu(E_i)| \geq |\mu(E)|$$

$|\mu|$ est une mesure positive sur $\mathcal{M}$ et est toujours de masse finie. De plus, il existe $h : X \to \mathbb{C}$ mesurable, telle que

$$|h(x)| = 1, \quad \forall x \in X \quad \text{et} \quad d\mu = hd|\mu| \quad \text{i.e.} \quad \int f d\mu = \int f h d|\mu|$$

pour toute fonction f continue et bornée.

théorème 30. représentation de Riesz-Markov : *Soient X un espace topologique séparé localement compact et une forme linéaire complexe $L \in C_0(X)$ sur $(C_0(X), \|\cdot\|_\infty)$. Alors il existe une mesure de Borel complexe et régulière μ qui représente L au sens où*

$$\forall f \in C_0(X) \qquad L(f) = \int_X f d\mu$$

De plus,

$$\|L\| = \sup_{\|f\|_\infty \leq 1} L(f) = |\mu|(X)$$

On appelle $|\mu|(X)$ la variation totale.

théorème 31. Cybenko (1989) : *Soit $\sigma : \mathbb{R} \to \mathbb{R}$ continue et sigmoïdale ($\lim_{x\to-\infty} \sigma(x) = 0$ et $\lim_{x\to\infty} \sigma(x) = 1$). Alors l'ensemble $\mathcal{N}_1$ des réseaux de neurones feedfoward à 1 couche cachée est dense dans $C([0;1]^d, \mathbb{R})$.*

Démonstration. En dimension $d = 1$, on va montrer le résultat dans des cas particuliers qui sortent du cadre du théorème. Dans le cas σ l'indicatrice de $\mathbb{R}_+$ (heaviside). On a vu que toute fonction f continue peut être approximée par une fonction constante par morceaux. Or toute fonction constante par morceaux est un réseaux de $\mathcal{N}_1$ d'activation heaviside. En effet pour l'intervalle $[a; b[$, et g qui vaut λ sur cet intervalle, on a

$$g(x) = \lambda(\sigma(x - a) - \sigma(x - b))$$

Pour la fonction partie positive (ReLU). On fait sensiblement la même chose en approchant par une fonction affine par morceaux.
Prouvons le théorème. On a $\mathcal{N}_1 \subset C([0;1]^d, \mathbb{R})$ et on veut montrer $\overline{\mathcal{N}_1} = C([0;1]^d, \mathbb{R})$. La preuve n'est pas constructive et procède par contradiction.
Puisque σ est continue, on a $\mathcal{N}_1 \subset \overline{\mathcal{N}_1} \subset C([0;1]^d, \mathbb{R})$. Supposons, par l'absurde, qu'il existe $f_0 \in C([0;1]^d, \mathbb{R}) \backslash \overline{\mathcal{N}_1}$. $C([0;1]^d, \mathbb{R})$ muni de la norme infini, est un espace vectoriel normé, on peut donc appliquer le théorème de séparation de Hahn-Banach avec $M = \overline{\mathcal{N}_1}$. Il existe donc une forme linéaire continue L telle que

$$L(f_0) = 1 \qquad \text{et} \qquad \overline{\mathcal{N}_1} \subset L^{-1}(\{0\})$$

On peut également appliquer le théorème de représentation de Riesz-Markov puisque $[0;1]^d$ est compact et donc $C([0;1]^d, \mathbb{R}) = C_0([0;1]^d, \mathbb{R})$. Il existe une mesure régulière

signée μ qui représente L. On applique donc cela à $f(x) = \sigma(\langle w; x\rangle + b)$ et on obtient donc pour tout $w \in \mathbb{R}^d$ et tout $b \in \mathbb{R}$

$$L(f) = \int_{[0,1]^d} \sigma(\langle w; x\rangle + b) d\mu = 0$$

On va maintenant montrer que $\mu = 0$. On fixe un $w \in \mathbb{R}^d$ et trois scalaires $(b, \lambda, \phi) \in \mathbb{R}^3$. Alors pour tout $x \in \mathbb{R}^d$ on pose

$$\sigma_\lambda(x) = \sigma(\lambda(\langle w; x\rangle + b) + \phi)$$

on a donc

$$\lim_{\lambda\to\infty} \sigma_\lambda(x) = \begin{cases} 1 & \text{si } \langle w; x\rangle + b > 0 \\ \sigma(\phi) & \text{si } \langle w; x\rangle + b = 0 \\ 0 & \text{si } \langle w; x\rangle + b < 0 \end{cases}$$

Or $\sigma_\lambda \in \mathcal{N}_1$ donc

$$\int_{[0,1]^d} \sigma_\lambda(x) d\mu(x) = 0 = \int_{[0,1]^d} \sigma_\lambda(x) h(x) d|\mu|(x)$$

En faisant, tendre λ vers l'infini, on a, par théorème de convergence dominée (car σ est continue et sigmoïdale donc bornée),

$$0 = \int_{[0,1]^d} \sigma_\infty(x) d\mu(x) = \mu(\pi_{w,b}) + \sigma(\phi)\mu(H_{w,b})$$

où $\pi_{w,b}$ est l'ensemble des x vérifiant $\langle w; x\rangle + b > 0$ et $H_{w,b}$ est l'ensemble des x vérifiant $\langle w; x\rangle + b = 0$. On peut donc faire tendre ϕ vers $-\infty$ et ∞ et on en déduit, puisque σ est sigmoïdale, que $\mu(\pi_{w,b}) = \mu(H_{w,b}) = 0$. Or μ est signée et donc ne peut pas conclure tout de suite que $\mu = 0$.
Pour tout $h : [-\|w\|_1; \|w\|_1] \to \mathbb{C}$ bornée, on pose

$$\Psi(h) = \int_{[0,1]^d} h(\langle w; x\rangle) d\mu(x)$$

On remarque que Ψ est linéaire et continue car $\Psi(h) \leq \|h\|_\infty |\mu|([0,1]^d)$. Par ailleurs, pour $h = \mathbf{1}_{[\theta;\infty[}$ on a $\Psi(h) = \mu(\pi_{w,-\theta}) + \mu(H_{w,-\theta}) = 0$. De même, $\Psi(\mathbf{1}_{[\theta_1;\theta_2]}) = 0$. Donc pour toute fonction en escalier h, par linéarité on a $\Psi(h) = 0$. Or on peut approcher toute fonction continue par une suite de fonction en escalier et donc pour toute fonction continue h, par continuité de Ψ, on a $\Psi(h) = 0$. En particulier, pour $h(t) = e^{it}$ on a

$$\forall w \in \mathbb{R}^d, \quad \int_{[0,1]^d} e^{i\langle w;x\rangle} d\mu(x) = 0$$

On reconnaît $\hat{\mu}$ la transformée de Fourier (ou inverse selon la convention) de μ. On admet qu'il suffit, pour montrer que $\mu = 0$, de montrer que pour tout f de la forme $C_0^\infty(\mathbb{R}^d, \mathbb{R})$ (les fonctions C^∞ à support compact)

$$\int_{[0,1]^d} f d\mu = 0$$

En effet, soit $f \in C_0^\infty(\mathbb{R}^d, \mathbb{R})$, pour tout $x \in [0,1]^d$,

$$f(x) = \int_{\mathbb{R}^d} \hat{f}(w) e^{i\langle w;x\rangle} dw$$

La formulation d'inversion de la transformée de Fourier est valide car f appartient à l'espace de Schwartz. On a alors

$$\int_{[0,1]^d} f(x) d\mu(x) = \int_{[0,1]^d} \int_{\mathbb{R}^d} \hat{f}(w) e^{i\langle w;x\rangle} dw d\mu(x)$$

et par Fubini,

$$\int_{[0,1]^d} f(x) d\mu(x) = \int_{\mathbb{R}^d} \hat{f}(w) \int_{[0,1]^d} e^{i\langle w;x\rangle} d\mu(x) dw = 0$$

Et donc on a bien $\mu = 0$. Et donc on a la contradiction

$$L(f_0) = 0 = 1$$

□

5.3 Extensions

Il existe plusieurs extensions de ce résultat. Pour des fonctions d'activation plus variées

théorème 32. Hornik (1991) : *Soit $K \subset \mathbb{R}^d$ un compact. Supposons que $\sigma : \mathbb{R} \to \mathbb{R}$ est continue, bornée et non-constante. Alors, l'ensemble des réseaux de neurones feeedforward à 1 couche cachée est dense dans $C(K, \mathbb{R})$.*

théorème 33. Hornik (1991) : *Soit μ une mesure de Borel positive sur $\mathbb{R}^d$, de masse finie. Supposons que $\sigma : \mathbb{R} \to \mathbb{R}$ est bornée et non-constante. Alors, l'ensemble des réseaux de neurones feeedforward à 1 couche cachée est dense dans $L^p(\mathbb{R}^d, \mathbb{R}, \mu)$ pour tout $1 \leq p < \infty$.*

théorème 34. Barron (1993) : *Soit $\sigma : \mathbb{R} \to \mathbb{R}$ une fonction sigmoïdale. Pour toute $f : \mathbb{R}^d \to \mathbb{R}$ admettant une représentation de Fourier, tout $r > 0$ et toute mesure de probabilité μ sur B_r, il existe un réseau de neurone g_N feedforward biaisé tel que*

$$\int_{B_r} (f - g_N)^2 d\mu \leq \frac{(2rC_f)^2}{N}$$

avec $C_f = \int_{\mathbb{R}^d} \|w\|_2 |\hat{f}(w)| dw$.

Ainsi une seule couche cachée est suffisante pour faire des approximations aussi précises que l'on veut.

6 Impact de la profondeur sur l'erreur de généralisation

Nous allons partiellement répondre à la question suivante l'expressivité des réseaux de neurones feedforward gagnent-ils à être plus profond à nombre de poids égal. Nous allons regarder un problème de classification binaire. Soit un échantillon $(X_i, Y_i)_{[\![1;n]\!]} \overset{\text{i.i.d.}}{\sim} P_{X,Y}$ sur $\mathcal{X} \times \{0,1\}$. Le but est de construire $\hat{h}_n : \mathcal{X} \to \{0,1\}$ (une fonction aléatoire au même titre que le mouvement Brownien) qui dépend de l'échantillon précédent telle que

$$R(\hat{h}_n) = \int_{\mathcal{X}} \mathbf{1}_{y \neq \hat{h}_n(x)} dP_{X,Y}(x,y) = \mathbb{P}(Y \neq \hat{h}_n(X) | (X_i, Y_i)_{[\![1;n]\!]})$$

soit le plus petit possible. On peut aussi demander (plus faible) de minimiser

$$\mathbb{P}(Y \neq \hat{h}_n(X)) = \mathbb{E}[R(\hat{h}_n)]$$

On va étudier des classificateurs $\hat{h}_n$ qui s'obtiennent à partir d'un réseau feedforward, de la forme

$$\hat{h}_n(x) = \mathbf{1}_{\hat{f}_n(x)>0} = \text{sgn}(\hat{f}_n(x))$$

où $\hat{f}_n$ est lui même un réseau feedforward. Essentiellement on a rajouté une activation heaviside. On va fixer une architecture de réseau et faire ne faire varier que les poids. Ceci définit une classe de fonction $\mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ et $\mathcal{H} = \text{sgn}(\mathcal{F}) \subset \mathbb{R}^{\mathcal{X}}$. On appelle **excès de risque**

$$R(\hat{h}_n) - \inf_{h \in \mathcal{H}} R(h)$$

où $\inf_{h\in\mathcal{H}} R(h)$ est un oracle. On ne peut pas calculer explicitement l'oracle car on n'a pas la loi jointe.

6.1 VC-dimension et borne de risque

Pour plus de détails, voir "fondement mathématique de l'apprentissage statistique" Christophe Giraud [8]. On va définir la **VC-dimension**, pour cela on se donne un ensemble de classificateurs $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$. On appelle **coefficient d'éclatement** (en anglais : **shattering coefficient - growth function**) la quantité suivante

$$\pi_{\mathcal{H}}(m) = \max_{(x_i)_{[\![1;m]\!]} \in \mathcal{X}^m} \# \left\{ (h(x_i))_{[\![1;m]\!]} \in \{0,1\}^m \middle| h \in \mathcal{H} \right\}$$

remarquons que $\pi_H(m) \leq 2^m$ et $\pi_H(m) \leq \#\mathcal{H}$ si $\mathcal{H}$ est finie. On appelle **dimension de Vapnik-Chervonenkis** de $\mathcal{H}$, notée $\text{VC}_{\text{dim}}(\mathcal{H})$ la quantité

$$\text{VC}_{\text{dim}}(\mathcal{H}) = \sup \left\{ m \in \mathbb{N} \middle| \pi_{\mathcal{H}}(m) = 2^m \right\}$$

avec $\pi_{\mathcal{H}}(0) = 1$. Remarquons que $\pi_{\mathcal{H}}(m) = 2^m$ c'est équivalent à l'existence de $(x_i)_{[\![1;m]\!]}$ tel que

$$\left\{ (h(x_i))_{[\![1;m]\!]} \in \{0,1\}^m \middle| h \in \mathcal{H} \right\} = \{0,1\}^m$$

ou encore, pour tout $\sigma \in \{0,1\}^m$ il existe $h \in \mathcal{H}$ tel que

$$\forall i \in [\![1;m]\!], \quad h(x_i) = \sigma_i$$

Ainsi, la VC-dimension est la taille m du plus grand échantillon que $\mathcal{H}$ peut éclater (i.e. au quel on peut assigner une étiquette de n'importe quel signe possible en appliquant des fonctions $h \in \mathcal{H}$ à l'échantillon). C'est une *mesure de complexité* de $\mathcal{H}$. L'exemple classique consiste à prendre

$$\mathcal{H} = \left\{ x \in \mathbb{R}^d \to \text{sgn}(\langle w, x\rangle + b) \middle| w \in \mathbb{R}^d, b \in \mathbb{R} \right\}$$

l'ensemble des perceptrons avec heaviside. On a dans ce cas $\text{VC}_{\dim}(\mathcal{H}) = d+1$. Pour voir cela, nous pouvons faire un schéma en traçant des hyperplans séparateurs. Pour plus de détails voir la preuve du début du cours pour le cas des réseaux ReLU.

Lemma 35. lemme de Sauer : *Soit $\mathcal{H}$ telle que $0 < V = VC_{dim}(\mathcal{H}) < \infty$. On a alors*

$$\pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{V} \binom{m}{i} \begin{cases} = 2^m & si\ m \leq V \\ \leq \left(\frac{me}{V}\right)^V & sinon \end{cases}$$

La preuve se fait récurrence et n'est pas simple.

Proposition 36. majoration du risque de l'ERM : *Soit $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$ telle que $V = VC_{dim}(\mathcal{H}) \in \mathbb{N}^*$. Alors*

$$\mathbb{E}_{(X_i,Y_i)_{[\![1;m]\!]}} \left[\sup_{h\in\mathcal{H}} \left| \mathbb{P}(Y \neq h(X)) - \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{Y_i \neq h(X_i)} \right| \right] \leq 2\sqrt{\frac{2\log(2\pi_{\mathcal{H}}(m))}{m}}$$

cela contrôle les déviations uniformes du risque empirique autour du vrai risque. En conséquence, l'ERM (empirical risk minimizer) $\hat{h}_m \in \arg\min_{h\in\mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{Y_i \neq h(X_i)}$ vérifie

$$\mathbb{P}(Y - \hat{h}_n(X)) - \inf_{h\in\mathcal{H}} \mathbb{P}(Y \neq h(X)) \leq 4\sqrt{\frac{2\log(2\pi_{\mathcal{H}}(m))}{m}}$$

On remarque, en utilisant le lemme de Sauer que pour tout $m \geq V$,

$$\log(\pi_{\mathcal{H}}(m)) \leq V \log\left(\frac{em}{V}\right)$$

ce qui donne une borne d'excès de risque en $\sqrt{\frac{V\log(em/V)}{m}}$. Notons que l'on s'affranchir du log avec des outils de "chaînage". En combinant ce qui précède avec l'inégalité de Mc Daimid on peut prouver la borne en grande proba de la forme

$$R(\hat{h}_n) - \inf_{h\in\mathcal{H}} \leq C\sqrt{\frac{V\log(em/V)}{m}} + \frac{1}{m}\log\left(\frac{1}{\delta}\right)$$

avec probabilité supérieure à $1-\delta$.

Proposition 37. minoration de l'excès de risque dans le pire cas : *Soit $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$ telle que $V = VC_{dim}(\mathcal{H}) \in \mathbb{N}^*$. Alors pour tout $m \leq c_1 V$*

$$\inf_{\hat{h}_m} \sup_{P_{X,Y} \in \mathcal{M}_1^+(\mathcal{X} \times \{0,1\})} \left\{ \mathbb{P}(Y \neq \hat{h}_m(X)) - \inf_{h \in \mathcal{H}} \mathbb{P}(Y \neq h(X)) \right\} \geq c_2 \sqrt{\frac{V}{m}}$$

ici, c_1 et c_2 sont des constantes absolues.

Cette proposition signifie que quelque soit le $\hat{h}_m$ il existe une loi jointe $P_{X,Y}$ qui rend l'excès de risque moyen de $\hat{h}_m$ au moins de l'ordre de $\sqrt{\frac{V}{M}}$. C'est une borne inférieure minimax. On a donc encadré notre minimiseur du risque empirique par des objets de même ordre de grandeur (si on parvient à retirer le log).

6.2 Contrôle de la VC-dimension

Pour plus de détails voir "Nearly tight VC-dimension and pseudodimension bounds for piecewise linear neural networks" Bartlett COLT 2017 [2] (Conference On Learning Theory). C'est une mise-à-jour de résultats plus anciens (même l'article est récent). On va majorer la VC-dimension d'un réseau feedforward (à architecture fixée) en fonction de

- L le nombre de couches (layers)
- U le nombre de neurones (computation units)
- W pour le nombre de poids (weights)

L'architecture du réseau est définie par un graphe orienté acyclique ayant un unique nœud de sortie. On autorise les connections entre couches non-consécutives (type ResNet). (*le théorème qui vient est long mais contient surtout des définitions*)

théorème 38. *On suppose la fonction d'activation $\sigma : \mathbb{R} \to \mathbb{R}$ polynomiale par morceaux $I_1, ..., I_{p+1}$ de degré majoré par d. Soient $L \geq 1$, $U \geq 3$, $d \geq 0$, $p \geq 1$ et $W \geq U \geq L$. Soit un réseau feedforward avec W paramètres, U neurones et L couches tel que décrit précédemment. On note k_i le nombre de neurones sur la i-ème couche. On suppose que les neurones ont pour fonction d'activation σ. En sortie la fonction d'activation est l'identité. On pose pour $i \in [\![1; L]\!]$,*

- *Si $d = 0$, W_i = nombre de paramètres utiles au neurone de la couche i = nombre d'arc entrant sur la couche $i + k_i$*
- *Si $d \geq 1$, W_i = nombre de paramètres utiles au neurone de la couche 1 à i.*

On pose

$$\overline{L} = \frac{1}{W} \sum_{i=1}^{L} W_i \in [1; L]$$

ça vaut 1 si $d = 0$ et c'est proche de L si les neurones sont concentrés sur les premières couches. On pose

$$R = \sum_{i=1}^{L} k_i (1 + (i-1)d^{i-1}) \leq U + U(L-1)d^{L-1}$$

ça vaut U pour $d = 0$ et on peut majorer par ULd^{L-1}. Alors la classe $\mathcal{F}$ de toutes les fonctions $f_{a\in\mathbb{R}^W} : \mathbb{R}^{entrée} \to \mathbb{R}$ vérifie pour tout $m \geq W$,

$$\pi_{sgn(\mathcal{F})}(m) \leq \prod_{i=1}^{L} 2\left(\frac{2emk_ip(1+(i-1)d^{i-1})}{W_i}\right)^{W_i} \qquad (1)$$

$$\leq \left(4emp(1+(i-1)d^{i-1})\right)^{\sum_{i=1}^{L} W_i} \qquad (2)$$

par ailleurs,

$$VC_{dim}(sgn(\mathcal{F})) \leq L+\overline{L}W\log_2(4epR\log_2(2epR)) = O(\overline{L}W\log(pU)+\overline{L}LW\log(d)) \qquad (3)$$

$\log_2(2epR)$ *est négligeable. En particulier,*

- *pour $d = 0$, $VC_{dim}(sgn(\mathcal{F})) \leq L + W\log_2(4epU\log_2(2epU)) = O(W\log(pW))$*
- *pour $d = 1$, $VC_{dim}(sgn(\mathcal{F})) = O(\overline{L}W\log(pU))$*

Ajoutons, pour $d = 1$, avec $W \geq cL$ et $L \geq c$ il existe un réseau ReLU à moins de L couches et moins de W paramètres vérifie

$$\mathrm{VC}_{\mathrm{dim}}(\mathrm{sgn}(\mathcal{F})) \geq \frac{WL}{c}\log\left(\frac{W}{L}\right)$$

Ainsi dans le pire des cas (en terme d'architecture à L, W fixés) un réseau ReLU est de l'ordre de $LW\log(W)$. On peut également faire la remarque "historique" suivante : $O(\overline{L}W\log(pU))$ est une meilleur majoration que $O(\min(W^2, WL\log(W) + L^2W))$ qui été la référence jusqu'alors.

Démonstration. La preuve repose sur le lemme suivant (c'est le résultat difficile de géométrie algébrique que l'on admet) voir "Neural Network Learning : Theoretical Foundations" de Anthony et Bartlett [1].

Lemma 39. *Soit $p_1, ..., p_m$ polynôme en n variables avec $n \leq m$.*

$$K = \#\{(sgn(p_1(a)), ..., sgn(p_m(a)))|a \in \mathbb{R}^n\}$$

Alors $K \leq 2\left(\frac{2emd}{n}\right)^n$.

Notons $f(x, a)$ la sortie du réseau pour l'entrée $x \in \mathcal{X} = \mathbb{R}^{\text{entrée}}$ et le vecteur de paramètres $a \in \mathbb{R}^W$. Soit un échantillon $(x_1, ..., x_m) \in \mathcal{X}^m$, afin de majorer $\pi_{\mathrm{sgn}(\mathcal{F})}(m)$, majorons

$$\#\left\{(\mathrm{sgn}(f(x_1, a)), ..., \mathrm{sgn}(f(x_m, a)))\middle| a \in \mathbb{R}^W\right\}$$

Or on ne peut aps appliquer le lemme immédiatement car la sortie du réseau est polynomiale par morceaux. On partitionner l'espace pour pouvoir appliquer le lemme.

$$\#\left\{(\mathrm{sgn}(f(x_1, a)), ..., \mathrm{sgn}(f(x_m, a)))\middle| a \in \mathbb{R}^W\right\} \leq \sum_{i=1}^{N} \#\left\{(\mathrm{sgn}(f(x_1, a)), ..., \mathrm{sgn}(f(x_m, a)))\middle| a \in P_i\right\}$$

Tout l'exercice se réduit à la construction d'une bonne partition. On va la construire par récurrence. On va construire $\mathcal{S}_0, ..., \mathcal{S}_{L-1}$ partitions de $\mathbb{R}^W$ telles que

— les partitions sont emboîtées, i.e. chaque $S \in \mathcal{S}_i$ est une union de $S' \in \mathcal{S}_{i+1}$

— $\mathcal{S}_0 = \mathbb{R}^W$ et

$$\frac{\#\mathcal{S}_i}{\#\mathcal{S}_{i-1}} \leq 2\left(\frac{2emk_ip(1+(i-1)d^{i-1})}{W_i}\right)^{W_i}$$

— pour tout $i \in [\![0, ..., L_1]\!]$, tout $S \in \mathcal{S}_i$ tout $j \in [\![1; m]\!]$, la sortie d'un neurone de la couche i est une fonction polynomiale de W_i variables (les paramètres utiles aux neurones de la couche i) de $a \in S$, de degré inférieur ou égal à id^i.

On procède par récurrence. Dans le cas $i = 0$, on pose $\mathcal{S}_0 = \mathbb{R}^W$ et la sortie d'un neurone d'entrée est une fonction constante sur $a \in \mathcal{S}_0$.
Dans le cas $i \geq 1$, Supposons avoir construit des partitions emboîtées $\mathcal{S}_0, ..., \mathcal{S}_{i-1}$ vérifiant les pré-requis précédents. Construisons $\mathcal{S}_i$. On note $p_{h,x_j,S}(a)$ l'entrée du h-ième neurone de la couche i, pour l'entrée x_j comme fonction de $a \in S$ avec $S \in \mathcal{S}_{i-1}$. D'après l'hypothèse de récurrence (c), puisque $p_{h,x_j,S}(a)$ est de la forme

$$\sum_k w_k \text{sortie}(\text{neurone}_k) + b$$

et puisque les partitions sont emboîtées, on a $p_{h,x_j,S}$ est polynomiale sur S, de degré inférieur ou égal à $1+(i-1)d^{i-1}$ et dépend d'au plus W_i variables. Cependant, à cause de σ la sortie $\sigma(p_{h,x_j,S}(\cdot))$ du neurone h est polynomiale par morceaux sur S. On va désintégrer S en sous-cellules pour que les sorties soient polynomiales sur ces morceaux. Soient $t_1, ..., t_p$ les coupures des intervalles $I_1, ..., I_{p+1}$. Considérons les polynômes $(p_{h,x_j,S}(a) - t_r)$ pour tous les h, j, r. D'après le lemme (on peut l'appliquer car $m \geq W \geq W_i$), cet ensemble de polynômes sur $\mathbb{R}^W$ atteint au plus

$$\pi = 2(2e(k_imp)(1+(i-1)d^{i-1})/W_i)_i^W$$

vecteurs de signes différents, avec k_imp le nombre de polynômes, et $1+(i-1)d^{i-1}$ le degré. On peut donc partitionner S en au plus π morceaux de sorte que sur chaque morceau, les $p_{h,x_j,S}$ sont polynomiaux. On a donc

$$\frac{\#\mathcal{S}_i}{\#\mathcal{S}_{i-1}} \leq \pi$$

Et de plus, le nouveau degré est majoré par $d(1+(i-1)d^{i-1}) \leq id^i$. Ceci clôt la récurrence. En particulier $\mathcal{S}_{L-1}$ est une partition de $\mathcal{R}^W$ telle que la sortie de chaque neurone de toute couche soit polynomiale de degré majoré par $(L-1)d^{L-1}$ sur chaque $S \in \mathcal{S}_{L-1}$. On applique le lemme sur chaque $S \in \mathcal{S}_{L-1}$ ce qui donne

$$\#\left\{(\text{sgn}(f(x_1,a)), ..., \text{sgn}(f(x_m,a)))\middle| a \in S\right\} \leq 2\left(\frac{2em(1+(L-1)d^{L-1})}{W_L}\right)^{W_L}$$

dès lors on conclut en injectant dans la majoration initiale.

$$\#\left\{(\text{sgn}(f(x_1,a)), ..., \text{sgn}(f(x_m,a)))\middle| a \in \mathbb{R}^W\right\} \leq \#\mathcal{S}_{L_1} \times 2\left(\frac{2em(1+(L-1)d^{L-1})}{W_L}\right)^{W_L}$$

On utilise la propriété de récurrence pour majorer le cardinal et on trouve le résultat (1)

$$\pi_{\text{sgn}(\mathcal{F})}(m) \leq \prod_{i=1}^{L} 2\left(\frac{2emk_ip(1+(i-1)d^{i-1})}{W_i}\right)^{W_i}$$

on utilise le fait que la moyenne géométrique est inférieur à la moyenne arithmétique. Ainsi,

$$\prod_{i=1}^{L} 2\left(\frac{2emk_ip(1+(i-1)d^{i-1})}{W_i}\right)^{W_i} \leq 2^L\left(\frac{2empR}{\sum_{i=1}^{L} W_i}\right)^{\sum_{i=1}^{L} W_i} \qquad (*)$$

Et après quelques majorations malines, on trouve le résultat (2)

$$\pi_{\text{sgn}(\mathcal{F})}(m) \leq \left(4emp(1+(i-1)d^{i-1})\right)^{\sum_{i=1}^{L} W_i}$$

On part de la majoration $(*)$ et on utilise le lemme

Lemma 40. *Soit $r \geq 16$ et $w \geq t > 0$. Alors pour tout $m > t + w\log_2(2r\log_2(r)) = x_0$, on a*

$$2^m > 2^t(mr/w)^w$$

Par définition de la VC-dimension et par la majoration $(*)$, et du lemme précédent avec $t = L$, $w = \sum_{i=1}^{L} W_i$ et $r = 2epR \geq 2eU \geq 16$ implique le résultat (3)

$$\text{VC}_{\text{dim}}(\text{sgn}(\mathcal{F})) \leq L + \overline{L}W\log_2(4epR\log_2(2epR)) = O(\overline{L}W\log(pU) + \overline{L}LW\log(d))$$

□

Donnons une preuve du dernier lemme.

Démonstration. On a pour tout $m > x_0$

$$2^m > 2^t\left(\frac{mr}{w}\right)^w \Leftrightarrow m - t - w\log_2\left(\frac{mr}{w}\right) > 0$$

On veut donc montrer que pour $m > x_0$ on a $f : m \to m - t - w\log_2\left(\frac{mr}{w}\right)$ qui est croissante et positive. On va donc montrer que $f(x_0) \geq 0$ et $f'(m) \geq 0$.

$$f(x_0) = x_0 - t - w\log_2\left(\frac{x_0r}{w}\right) = w\left(\log_2(2r\log_2(r)) - \log_2\left(\frac{x_0r}{w}\right)\right)$$

ainsi

$$f(x_0) \geq 0 \Leftrightarrow 2r\log_2(r) \geq \frac{x_0r}{w}$$

i.e.

$$f(x_0) \geq 0 \Leftrightarrow 2w\log_2(r) \geq x_0 = t + w\log_2(2r\log_2(r))$$

i.e.

$$f(x_0) \geq 0 \Leftrightarrow \log_2\left(\frac{r^2}{2r\log_2(r)}\right) \geq \frac{t}{w}$$

or $r \geq 16$ et donc

$$\frac{r^2}{2r\log_2(r)} \geq 2 > 2^{\frac{t}{w}}$$

On en déduit $f(x_0) \geq 0$. Il faut montrer $f'(m) \geq 0$ pour tout $m \geq x_0$.

$$f' : m \to 1 - \frac{w}{m\log(2)}$$

i.e.

$$\forall m \geq x_0, \quad f'(m) \geq 0 \Leftrightarrow \forall m \geq x_0, \quad 1 \geq \frac{w}{m\log(2)}$$

$$\forall m \geq x_0, \quad f'(m) \geq 0 \Leftrightarrow x_0 \geq \frac{w}{\log(2)}$$

Or $x_0 \geq w\log_2(2r\log_2(r))$, ainsi $x_0 \geq \frac{w}{\log(2)}$ car $r \geq 16$. □

7 Impact de la profondeur sur l'expressivité des réseaux feedforward

On peut donner une réponse partielle à la question : à nombre de poids W égal, les réseaux de neurones feedforward profonds (et donc étroits) sont-ils plus expressifs que les réseaux peu profonds (et donc larges) ? En anglais on les appelle **shallow**. Nous allons regarder un cas particulier. Celui de l'approximation dans le pire des cas de fonctions Lipschitz. Nous allons nous appuyer sur l'article suivant "Optimal Approximation of continious functions by very deep ReLU networks" COLT 2018 Yarotsky [20].

$$\mathcal{L}_1([0;1]^d) = \{f : [0;1]^d \to \mathbb{R} | \forall x, y, |f(x) - f(y)| \leq \|x - y\|\}$$

On reprend les notations précédentes. On représente un réseau comme un DAG (directed acyclic graph) $G = (A, S)$. On considère une fonction d'activation $\sigma : \mathbb{R} \to \mathbb{R}$ et $U - 1$ neurones cachés de la forme

$$x \to \sigma(\langle w, x\rangle + b)$$

Un nombre de couches L (profondeur du réseau), U neurones et W poids. Ainsi, on note $f_a : [0;1]^d \to \mathbb{R}$ l'action du réseau avec $a \in \mathbb{R}^W$ les poids du réseau. On appellera **architecture** de réseau $\mathcal{A}$, la donnée de G et σ. À chaque architecture $\mathcal{A}$ correspond

$$\mathcal{F} = \left\{f_a \middle| a \in \mathbb{R}^W\right\} \text{ et } \mathcal{H} = \text{sgn}(\mathcal{F})$$

Enfin on notera $\text{VCdim}(\mathcal{A}) = \text{VCdim}(\mathcal{H})$.

7.1 Capacité d'approximation et VC-dimension

Intuitivement, un réseau avec une bonne capacité d'approximation doit être complexe et donc de VC-dimension élevée. Le nom du théorème est le suivant "Error bounds for approximations of deep ReLU networks"

théorème 41. Yarotsky 2017 théorème 4(a) : *Soit $d \in \mathbb{N}^*$. Il existe $\epsilon_d \in]0;1[$ et $c_d > 0$ tels que il existe $\epsilon \in]0;\epsilon_d[$ et $\mathcal{A}$ une architecture de réseau de neurones profond à W poids capable d'approcher toute fonction $f \in \mathcal{L}_1([0;1]^d)$ à ϵ près. Alors*

$$VCdim(\mathcal{A}) \geq c_d \left(\frac{1}{\epsilon}\right)^d \qquad (1)$$

Par conséquent, la pire erreur d'approximation par $\mathcal{A}$ sur $\mathcal{L}_1([0;1]^d)$ vérifie

$$\sup_{f \in \mathcal{L}_1([0;1]^d)} \inf_{a \in \mathbb{R}^W} \|f_a - f\|_\infty \geq \left(\frac{c_d}{\text{VCdim}(\mathcal{A})}\right)^{\frac{1}{d}} \wedge \epsilon_d \qquad (2)$$

Démonstration. On a

$$\text{VCdim}(\mathcal{A}) = \sup\{m \in \mathbb{N}^* | \pi_H(m) = 2^m\}$$

$$= \sup\left\{m \in \mathbb{N}^* | \exists x_1, ..., x_m \text{qui est pulvérisé par les } \mathbf{1}_{f_a > 0}, a \in \mathbb{R}^W\right\}$$

Exhibons un échantillon de taille $m \simeq (1/\epsilon)^d$ pulvérisé par $H = \{\mathbf{1}_{f_a>0}, a \in \mathbb{R}^W\}$ (voir warm-up théorème 25 et 26). Dans l'hypercube $[0;1]^d$ on subdivise chaque coordonnées en N intervalles, on considère $m = (N+1)^d$ points $x_1, ..., x_m$ qui vérifient

$$\forall i \neq j \quad \|x_i - x_j\| \geq \frac{1}{N}$$

Soit $\phi : \mathbb{R}^d \to \mathbb{R}$ de classe C^∞ telle que $\phi(0) = 1$ et $\|x\| > 0.5 \Rightarrow \phi(x) = 0$. Pour $y_1, ..., y_m \in \mathbb{R}$ choisis ultérieurement, on définit $f : [0;1]^d \to \mathbb{R}$ par

$$f(x) = \sum_{i=1}^m y_i \phi(N(x - x_i))$$

On constate que f est de classe C^∞ et $f(x_i) = y_i$ pour tout $i \in [\![1;m]\!]$. Les $\phi(N(x-x_i)$ sont à support 2 à 2 disjoints et de même pour leurs dérivées partielles. Il faut maintenant vérifier que $f \in \mathcal{L}_1([0;1]^d)$. Soit $x \in [0;1]^d$ et $j \in [\![1;m]\!]$

$$\partial_j f(x) = \sum_{i=1}^m y_i N \partial_j \phi(N(x - x_i))$$

Il existe i dépendant de x et non de j tel que

$$\partial_j f(x) = y_i N \partial_j \phi(N(x - x_i))$$

ainsi

$$|\nabla f(x)| = y_i N \nabla \phi(N(x - x_i))$$

et donc

$$\|\nabla f(x)\| \leq y_i N \sup_{u \in \mathbb{R}} \nabla \phi(x) = |y_i| N l$$

Si on choisit les $|y_i| \leq 1/Nl$ on a alors $\|\nabla f\| \leq 1$. On prend N la partie entière de $1/2l\epsilon$ dès que $\epsilon \leq 1/2l$ et $N \leq 1/2l\epsilon$ sinon. Par hypothèse, $\mathcal{A}$ peut approcher f à ϵ près et donc à $1/2lN$ près. D'où, il existe $a \in \mathbb{R}^W$ tel que

$$\forall i \in [\![1; m]\!], \quad |f_a(x_i) - f(x_i)| \leq \frac{1}{2lN}$$

ainsi

$$f_a(x_i) \begin{cases} > 0 & \text{si } z_i = 1 \\ < 0 & \text{sinon} \end{cases}$$

Le vecteur z étant quelconque dans $\{-1, 1\}^m$, on vient de montrer que $(x_1, ..., x_m)$ est pulvérisé par $\{\mathbf{1}_{f_a(x)>0}, a \in \mathbb{R}^W\} = H$ □

7.2 Capacité d'approximation pour σ constante par morceaux

théorème 42. *Soit $d \in \mathbb{N}^*$, il existe $b_d > 0$ (interprétée comme potentiellement grande) et $b'_d > 0$ (interprétée comme potentiellement petite) telles que :*

- *pour tout $W \geq b_d$ et toute architecture $\mathcal{A}$ à W poids et fonction d'activation σ constante par morceaux à au moins $p + 1 \geq 2$ morceaux, on a*

$$\sup_{f \in \mathcal{L}_1([0;1]^d)} \inf_{a \in \mathbb{R}^W} \|f_a - f\|_\infty \geq b'_d (W \ln(pW))^{-\frac{1}{d}}$$

- *pour tout $W \geq b_d$ il existe une architecture $\mathcal{A}$ à 2 couches cachées et moins de W poids de fonction d'activation $\mathbf{1}_{x \geq 0}$, telle que*

$$\sup_{f \in \mathcal{L}_1([0;1]^d)} \inf_{a \in \mathbb{R}^W} \|f_a - f\|_\infty \leq b_d W^{-\frac{1}{d}}$$

Démonstration. Pour montrer le premier point, on applique l'inégalité (2) du théorème de la sous-section précédente.

$$\sup_{f \in \mathcal{L}_1([0;1]^d)} \inf_{a \in \mathbb{R}^W} \|f_a - f\|_\infty \geq \left(\frac{c_d}{\text{VCdim}(\mathcal{A})} \right)^{\frac{1}{d}} \wedge \epsilon_d$$

Or, d'après ce qui précède (résultat sur la VC-dimension), pour $U \geq 3$ (ce qui est vrai dès que $W : geq 2d + 4$), par le théorème 37

$$\text{VCdim}(\mathcal{A}) \leq C_d W \log(pW)$$

d'où

$$\sup_{f \in \mathcal{L}_1([0;1]^d)} \inf_{a \in \mathbb{R}^W} \|f_a - f\|_\infty \geq \left(\frac{c_d}{C_d W \log(pW)} \right)^{\frac{1}{d}} \wedge \epsilon_d \geq b'_d (W \ln(pW))^{-\frac{1}{d}}$$

Ce qui montre le premier point. Montrons le second. On va approcher tout $f \in \mathcal{L}_1([0; 1]^d)$ par une fonction constante par morceaux. On subdivise l'hypercube $[0; 1]^d$ en N^d cubes

de longueur $1/N$. On note x_i le centre du $i^{\text{ème}}$ cube C_i avec $i \in [\![1; N^d]\!]$. Les C_i sont deux à deux disjoints. La fonction constante par morceaux $\tilde{f}$ est définit comme suit

$$\tilde{f} : x \to \sum_{i=1}^{N^d} f(x_i)\mathbf{1}_{x \in C_i}$$

Puisque $f \in \mathcal{L}_1([0;1]^d)$ on a pour tout $x \in [0;1]^d$ on a

$$|f(x) - \tilde{f}(x)| \leq \frac{\sqrt{d}}{2N}$$

d'où

$$\|f - \tilde{f}\|_\infty \leq b_d W^{-\frac{1}{d}}$$

dès que $N \simeq W^{\frac{1}{d}}$. Il reste à montrer que $\tilde{f}$ peut être implémentée par un réseau heaviside à deux cachées. Pour chaque cube C_i et chacune de ses $2d$ faces, on peut coder le fait d'être du bon coté de la face via un perceptron $x \to \mathbf{1}_{\langle w_j; x\rangle + b \geq 0}$ pour $j \in [\![1; 2d]\!]$. Ainsi appartenir au cube signifie

$$\mathbf{1}_{C_i} = \mathbf{1}_{\sum_{j=1}^{2d} \mathbf{1}_{\langle w_j^i; x\rangle + b^i \geq 0} = 2d} = \sigma\left(\sum_{j=1}^{2d} \sigma(\langle w_j^i; x\rangle + b^i) - 2d\right)$$

et donc

$$\tilde{f}(x) = \sum_{i=1}^{N^d} f(x_i)\sigma\left(\sum_{j=1}^{2d} g(\sigma(\langle w_j^i; x\rangle + b^i)) - 2d\right)$$

avec g l'identité si la face appartient au cube et sinon $g(u(t)) = 1 - u(-t)$. Et donc $\tilde{f}$ est bien un réseau à deux couches cachées de fonction d'activation heaviside. Il reste à vérifier que le nombre de poids est inférieur à W. En effet, pour $N = \left\lfloor \left(\frac{W}{2+2d(d+2)}\right)^{\frac{1}{d}} \right\rfloor \geq 1$

$$\text{nb poids} = N^d + N^d(1 + 2d + 2d(1 + d)) = 2N^d(d^2 + 2d + 1) \leq W$$

□

7.3 Capacité d'approximation pour σ polynomiale par morceaux

En prenant σ polynomiale par morceaux de degré supérieur ou égal à 1 on améliore l'approximation, on passe à $W^{-2/d}$. Dans cette sous-section, on s'autorise des énoncés approximatifs. Typiquement les inégalités seront énoncés à multiplication par une constante près.

théorème 43. Yarotsky 2018 théorèmes 1 et 2 :

— *pour W assez grand et toute architecture à W poids, avec σ polynomiale par morceaux de degré ≥ 1,*

$$\sup_{f \in \mathcal{L}_1([0;1]^d)} \inf_{a \in \mathbb{R}^W} \|f_a - f\|_\infty \geq W^{-\frac{2}{d}}$$

— *pour W assez grand, il existe une architecture à au plus W poids, de fonction d'activation ReLU telle que*

$$\sup_{f\in\mathcal{L}_1([0;1]^d)}\inf_{a\in\mathbb{R}^W}\|f_a-f\|_\infty\leq W^{-\frac{2}{d}}$$

— *pour W assez grand et toute architecture à W poids et $\sigma=ReLU$, si on a*

$$\sup_{f\in\mathcal{L}_1([0;1]^d)}\inf_{a\in\mathbb{R}^W}\|f_a-f\|_\infty\leq W^{-\frac{2}{d}}$$

avec $k\in[1;2]$, alors l'architecture est de profondeur

$$L\geq\frac{W^{k-1}}{\log(W)}$$

Ainsi la meilleure approximation au sens du pire des cas sur $\mathcal{L}_1([0;1]^d)$ avec W poids est de l'ordre de $W^{-\frac{2}{d}}$, et elle est atteinte (dans le cas $\sigma=$ ReLU) que par une architecture de longueur $L\geq\frac{W^{k-1}}{\log(W)}$.

théorème 44. Devore et al 1989 théorème 4.2 : *Pour toute application $M:\mathbb{R}^W\to C([0;1]^d)$ et toute fonction $\bar{a}:\mathcal{L}_1([0;1]^d)\to\mathbb{R}^W$ continue pour la norme infinie, on a*

$$\sup_{f\in\mathcal{L}_1([0;1]^d)}\|M(\bar{a}(f))-f\|_\infty\geq W^{-\frac{1}{d}}$$

Ce théorème est cité dans l'article de Yarotsky. Si on décide de voir $M\circ\bar{a}$ comme une application qui associe le meilleur réseau ReLU, on a alors que le mapping $\bar{a}$ n'est pas continue en f si on veut des réseaux ayant les performances annoncées par le théorème de Yarotsky. Notons quand la preuve du théorème 41, la fonction de mapping est continue (même linéaire) en f.

Démonstration. Montrons les points 1 et 3 du théorème 42. On va utiliser le théorème 37 équation (3). On a alors

$$\mathrm{VC}_{\dim}(\mathcal{A})\leq\overline{L}W\log(pU)+\overline{L}LW\log(\deg)$$

i.e.

$$\mathrm{VC}_{\dim}(\mathcal{A})\leq LW\log(pU)+L^2W\log(\deg)$$

Comme $L\leq W$ on a

$$\mathrm{VC}_{\dim}(\mathcal{A})\leq W^2\log(pU)+W^3\log(\deg)$$

En fait il y a mieux d'après Goldberg et Jerrum, avec $\mathrm{VC}_{\dim}(\mathcal{A})\leq W^2$ (mais en pratique L est beaucoup plus petit que W et donc le théorème 37 n'est pas vide de sens). D'après le théorème 40 équation (2) on a maintenant

$$\sup_{f\in\mathcal{L}_1([0;1]^d)}\inf_{a\in\mathbb{R}^W}\|f_a-f\|_\infty\geq\left(\frac{1}{\mathrm{VC}_{\dim}(\mathcal{A})}\right)^{1/d}\wedge\epsilon_d\geq W^{-2/d}$$

pour W assez grand. Ce qui démontre le point 1. Par un raisonnement analogue en partant de la borne

$$\mathrm{VC}_{\dim}(\mathcal{A}) \leq LW\log(W)$$

pour une architecture ReLU, on trouve bien le point 3. Montrons maintenant le point 2

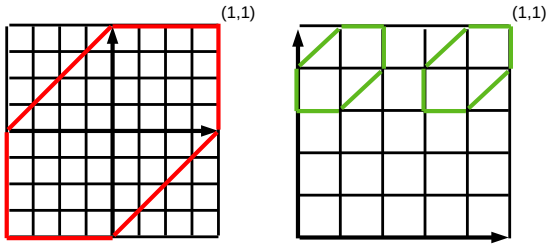

FIGURE 6 – Triangulation P_N pour $N = 1$ et $N = 5$

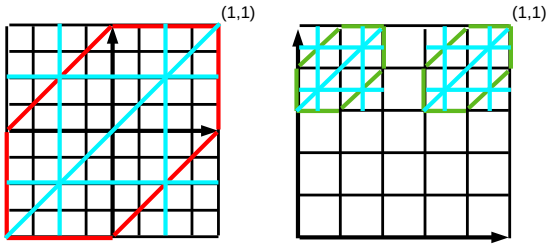

FIGURE 7 – Sous-triangulation P_M de P_N pour $N = 1$ et $N = 5$

du théorème 42. En préliminaires, on a

$$\forall x, y, \quad \max\{x, y\} = x + (y - x)_+ \text{et}\, \min\{x, y\} x - (x - y)_+$$

et

$$\forall x \in [0; 1], \forall y \in \{-1, 0, 1\}, \quad xy = (x + y - 1)_+ + (-x - y)_+ - (-y)_+$$

Ce dernier point est le point clef qui nous distingue du cas heaviside. On va suivre le schéma de construction suivant

- on va approcher f à $\epsilon \simeq W^{-1/d}$ par un réseau ReLU à $W/2$ poids et peu profond. Ainsi $\tilde{f}_1$ donne une approximation "grossière" de f. C'est une interpolation affine de f sur les sommets d'une triangulation.
- On pose $f_2 = f - \tilde{f}_1$. On approche f_2 à $\epsilon \simeq W^{-2/d}$ près par un réseau ReLU profond à $W/2$ poids et de profondeur $L \simeq W$. On construit ainsi $\tilde{f}_2$ en utilisant une technique de quantification.
- On conclut en posant $\tilde{f} = \tilde{f}_1 + \tilde{f}_2$.

On considère la triangulation P_N de $\mathbb{R}^d$ définie à l'aide des hyperplans d'équations $x_k - x_l = n/N$ et $x_k = n/N$ avec $n \in \mathbb{Z}$ et $k, l \in [\![1; d]\!]$. Voir fig 6. On définit

$$\phi : x \to \phi\left(N\left(x - \frac{\vec{n}}{N}\right)\right)$$

avec ϕ affine sur chaque simplexe de P_1. Elle est donc définit par sa valeur au sommets et on pose $\phi(0) = 1$ et ϕ vaut 0 sur les autres sommets. On pose alors

$$\tilde{f}_1(x) = \sum_{\vec{n} \in [\![1;N]\!]^d} f\left(\frac{\vec{n}}{N}\right) \phi(Nx - \vec{n})$$

$\phi(Nx - \vec{n})$ est encodable par un réseau ReLU dont la profondeur ne dépend que de d. C'est l'unique interpolation affine de f sur la triangulation. On peut montrer que $\tilde{f}_1$ est $\sqrt{d}$-Lipschitz et donc

$$\|\tilde{f}_1 - f\|_\infty \leq \frac{d}{N} \leq W^{-\frac{1}{d}}$$

avec $N \simeq W^{1/d}$. On prend maintenant $f_2 = f - \tilde{f}_1$ et on va considérer une sous-triangulation P_M plus fine que P_N (M multiple de N). Voir fig 7.

$$\forall \vec{q} \in S = \{0, 1, 2\}^d, \quad g_{\vec{q}} = \sum_{\vec{n} \in (\vec{q} + (3\mathbb{Z})^d) \cap [\![0;N]\!]^d} \phi(Nx - \vec{n})$$

$\vec{q}$ est un sommet de référence et on le décale de trois en trois et on reste dans le cube. On pose $f_{2,\vec{q}} = f_2 g_{\vec{q}}$. On remarque que

$$\sum_{\vec{n} \in [\![0;N]\!]^d} \phi(Nx - \vec{n}) = 1$$

et donc

$$f_2 = \sum_{\vec{q} \in S} f_{2,\vec{q}}$$

Il suffit d'approcher chaque $f_{2,\vec{q}}$ par $\tilde{f}_{2,\vec{q}}$. On pose $\lambda \simeq d^{3/2}/M$ tel que

$$\sup_{|x-y| \leq 1/M} |f_{2,\vec{q}}(x) - f_{2,\vec{q}}(y) \leq \lambda$$

ainsi

$$\tilde{f}_{2,\vec{q}}(\vec{m}/M) = \lambda \left\lfloor \frac{f_{2,\vec{q}}(\vec{m}/M)}{\lambda} \right\rfloor$$

On étend $\tilde{f}_{2,\vec{q}}$ à $[0; 1]^d$ par linéarisation sur chaque simplexe de P_M. On peut vérifier que

$$\|\tilde{f}_{2,\vec{q}} - f_{2,\vec{q}}\|_\infty \leq \frac{d^{S/2}}{M}$$

et donc en posant $\tilde{f} = \tilde{f}_2 + \tilde{f}_1$ on obtient

$$\|\tilde{f} - f\|_\infty \leq \|\tilde{f}_2 - f_2\|_\infty \leq \sum_{\vec{q} \in S} \|\tilde{f}_{2,\vec{q}} - f_{2,\vec{q}}\|_\infty \leq 3^d \frac{d^{S/2}}{M}$$

Et ainsi pour $M \simeq W^{2/d}$,

$$\|\tilde{f} - f\|_\infty \leq W^{-2/d}$$

Rappelons que $f_{2,\vec{q}}$ est supportée sur

$$\bigcup_{\vec{n} \in (\vec{q} + (3\mathbb{Z})^d) \cap [\![0;N]\!]^d} \times_{j=1}^d \left[\frac{n_j - 1}{N}; \frac{n_j + 1}{N} \right]$$

Des cubes deux à deux disjoints. **L'astuce consiste à coder intelligemment les valeurs**

$$\tilde{f}_{2,q} \left(\frac{\vec{n}}{N} + \frac{\vec{m}}{M} \right)$$

Pour $\vec{m}$ et $\vec{m}'$ voisins on code

$$\tilde{f}_{2,q} \left(\frac{\vec{n}}{N} + \frac{\vec{m}}{M} \right) - \tilde{f}_{2,q} \left(\frac{\vec{n}}{N} + \frac{\vec{m}'}{M} \right) \in \{-1, 0, 1\}$$

On a donc $3^{(2M/N-1)^d}$ possibilités et une écriture ternaire

$$b_{\vec{q},\vec{m}} = \sum_{t=1}^{(2M/N-1)^d} 3^t (B_{\vec{q},\vec{m}}(\vec{m}_t) + 1)$$

Les $b_{\vec{q},\vec{m}}$ seront des poids du réseau dont on peut extraire la valeur via l'opération $3\cdot - \lfloor 3\cdot \rfloor$ qui est approchable par un réseau ReLU. Ainsi le réseau obtenu a $N + (M/N)^d$ poids. A M fixé, on minimise le nombre de poids avec $N = \sqrt{M}$ et ainsi on a bien $M^{2/d}$ poids. □

Ce résultat est un résultat à la fois pessimiste et optimiste :

— optimiste : on se place dans le pire des cas avec un espace de fonction énorme.

— pessimiste : on ne dit pas comment optimiser le réseau. En pratique on ne connaît pas f or ici on utilise la connaissance de cette dernière.

On a donc une cible théorique dont on cherche à s'approcher avec un optimiseur. Notons tout de même que nous n'avons pas mis de bruit dans le problème considéré. L'idéal serait d'avoir une théorie unifiée. Dans le cas des RKHS se genre de résultats sont déjà connus.

8 Robustesse des Réseaux de neurones

On part du constat que les réseaux de neurones sont un modèle relativement ancien (1986 pour la backpropogation). Cependant, depuis 2010, on voit l'essor de ces modèles. La date clef est celle de 2012 avec la prouesse, faite sous la direction de Hinton, en classification d'image. Depuis la tendance n'a fait que croître. Ils ont impactés quasiment tous les champs d'application. Les deux architectures fondamentales pour faire fonctionner ces modèles sont

— les architectures convolutives, où un neurone est connecté qu'un sous-ensemble des informations d'entrée. On parle de sparse connectivity et de shared weights. Ceci permet de réduire énormément le nombre de paramètres.

— les architectures récurrentes, où la sortie du réseau est encodée en comme une mémoire pour la prochaine exécution.

Les modèles convolutifs sont particulièrement efficaces en Computer Vision. Les réseaux récurrents servent surtout à traiter les données qui présentent un caractère séquentiel. Dans le cas des réseaux récurrents, les problèmes de disparition du gradient sont très importants (beaucoup plus que pour les autres architectures). Les connexions résiduels forment une solution qui fut apportée à ce genre de problème.
Même si ces algorithmes marchent beaucoup, on a tout de même besoin de certifier ces systèmes. En français : "quand le modèle se trompe on veut qu'il ne se trompe de manière catastrophique". On va aborder les points suivants

— On va s'intéresser à la stabilité de la fonction de décision, ce qui peut être testé par attaque adversaire. On teste avec des données similaires et on teste, ainsi, la fiabilité de la fonction de décision. L'objectif de l'adversaire est de générer le plus petit masque possible permettant de donner une prédiction différente (voir même radicalement différente). On constate empiriquement que les réseaux profonds ne sont pas nécessairement très stables selon ce critère.

— On s'intéressera également à l'incertitude décisionnelle. On veut essentiellement savoir quand est-ce qu'on ne sait pas.

En général les réseaux de neurones font des prédictions sur-confiantes. Un modèle est dit "bien calibré" lorsque la confiance est égale à la précision effective du modèle. Et on peut donc se servir directement de cette confiance. Sinon on doit utiliser d'autres moyens.

8.1 Incertitude décisionnelle

On peut distinguer deux types décisionnelles

— L'incertitude aléatoire : cette incertitude vient des données. Le bruit est inhérent aux données. On ne peut pas la réduire mais on peut l'apprendre. On a par exemple : de la pluie, manque de features, la luminosité ou encore l'occlusion. On peut distinguer deux modèles :

  — **homoscedastic** : reste constante pour les données (incertitude moyenne)

  — **heteroscdastic** : dépend de chaque donnée

— L'incertitude épistémique, c'est l'incertitude qui vient du modèle prédictif. Cela correspond à la prédiction de données radicalement différentes des données d'entraînement.

Dans le cadre bayésien, cette incertitude épistémique est représentée comme une incertitude sur les valeurs des paramètres du modèle. On rappelle la formule fondamentale des statistiques bayésiennes. On a

$$\mathbb{P}(Y, w|X) = \mathbb{P}(Y|X, w)\mathbb{P}(w) = \mathbb{P}(w|X, Y)\mathbb{P}(Y|X)$$

or

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)}$$

i.e.

$$\mathbb{P}(w|X, Y) = \frac{\mathbb{P}(Y|w, X)\mathbb{P}(w)}{\mathbb{P}(Y|X)} \propto \mathbb{P}(Y|w, X)\mathbb{P}(w) \qquad (1)$$

Notons que les données seules n'apportent aucune information sur w et donc $\mathbb{P}(w|X) = \mathbb{P}(w)$. On modèle la vraisemblance $\mathbb{P}(Y|X, w)$, le prior $\mathbb{P}(w)$ et on calcule le posterior $\mathbb{P}(w|X, Y)$ comme $\mathbb{P}(Y|w, X)\mathbb{P}(w)$. On peut, au choix, faire

— le maximum de vraisemblance : trouver θ qui maximise $\mathbb{P}(Y|X, w)$. Cela suppose un prior uniforme.

— le maximum à posteriori : trouver w qui maximise $\mathbb{P}(w|X, Y)$.

Pour de nouvelles données x^*, y^*

$$p(y^*|x^*, Y, X) = \int p(y^*|x^*, w)p(w|X, Y)dw = \mathbb{E}_{p(w|D)}[p(y^*|x^*, w)] \qquad (2)$$

Ceci donne une mesure d'incertitude, via la concentration de la distribution ainsi prédite. En classification binaire, entraîner un réseau de neurones pour apprendre une fonction f à partir de données uni-dimensionnelles et on applique un softmax pour avoir des probabilités. Ceci donne une forte confiance pour des données éloignées des données d'entraînement, ce qui est injustifié. Cependant la limite du modèle bayésien est de deux ordres. Premièrement, en général, (1) n'a pas de forme explicite. Et (2) est compliqué à calculer (intégrale en grande dimension).

8.1.1 Régression linéaire bayésienne

On se donne N exemples d'entraînements (x_i, y_i) avec $y_i = w^T x_i + \epsilon_i$ avec ϵ_i i.i.d. normale centrale réduite. On a alors

$$p(y_i|x_i, w) \sim \mathcal{N}(w^T x_i, \sigma^2)$$

Dans le cas homoscedastic, σ est indépendant de x. En notation matricielle, on prend Φ de taille $N \times (p + 1)$

$$\Phi = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,p} \\ 1 & x_{2,1} & \dots & x_{2,p} \\ 1 & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & \dots & x_{N,p} \end{pmatrix}$$

On ajoute un 1 pour représenter le biais. On note Y le vecteur des labels. On a alors

$$Y = \Phi w + \epsilon$$

Quand on veut entraîner ce modèle avec un but de maximisation de la vraisemblance. On a

$$p(Y|X, w, \sigma) = \prod_{i=1}^{N} p(y_i|x_i, w, \sigma)$$

On a donc

$$\text{MLE} = \arg\min - \sum_{i=1}^{n} \log(p(y_i|x_i, w, \sigma))$$

i.e.

$$\text{MLE} = \arg\min - \sum_{i=1}^{n} (y_i - w^T \Phi_i)^2$$

On a alors pour solution

$$\hat{w} = (\Phi^T \Phi)^{-1} \Phi^T Y$$

Pour σ, on trouve

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^{n} (y_i - w^T \Phi_i)^2$$

Cependant, rappelons que la maximisation de la vraisemblance à des limites (typiquement lorsqu'on a peut de données). Dans ce cas, on préféra maximiser à partir d'un prior.
Si on prend comme prior $p(w|\alpha) = \mathcal{N}(w|0, \alpha^{-1}\text{Id})$. La vraisemblance devient, comme avant,

$$p(y_i|x_i, w) = \mathcal{N}(w^T \Phi_i, \beta^{-1}) \qquad \beta = \frac{1}{2\sigma^2}$$

Le posterior est lui aussi gaussien. On obtient

$$p(w|X, Y) = \mathcal{N}(w|\mu, \Sigma)$$

avec

$$\Sigma^{-1} = \alpha \text{Id} + \beta \Phi^T \Phi \text{ et } \mu = \beta \Sigma \Phi^T Y$$

Si on fait tendre α vers 0 on retrouve le maximum de vraisemblance et si on fait tendre N vers 0 on retrouve le prior. Remarquons qu'ajouter un prior gaussien de précision α sur les poids, agit comme un terme de régularisation l^2 de paramètre $\lambda = \alpha/\beta$.
On va donc calculer

$$p(y^*|x^*, D, \alpha, \beta) = \int p(y^*|x^*, w, \beta) p(w|D, \alpha, \beta) dw$$

C'est une convolution de gaussienne et c'est donc gaussien.

$$p(y^*|x^*, D, \alpha, \beta) = \mathcal{N}\left(y^* \middle| \mu^T \Phi(x^*), \frac{1}{\beta} + \Phi(x^*)^T \Sigma \Phi(x^*)\right)$$

$\Phi(x^*)^T \Sigma \Phi(x^*)$ est notre incertitude épistémique. β vient des données.

$$\sigma^2_{\text{pred}} = \frac{1}{\beta} + \Phi(x^*)^T \Sigma \Phi(x^*)$$

On a bien

$$\lim_{N \to \infty} \sigma^2_{\text{pred}} = 0$$

Dans le cas uni-dimensionnel on a

$$\Sigma^{-1} = \begin{pmatrix} \alpha \text{Id} + \beta N & \beta \mathbf{1}^T X \\ \beta \mathbf{1}^T X & \alpha \text{Id} + \beta X^T X \end{pmatrix}$$

Et, de plus, $\Phi(x^*)^T \Sigma \Phi(x^*)$ croit lorsque x est loin des exemples d'entraînement. On a un soucis lorsque les points sont espacés. En effet, dans ce cas, puisque le barycentre des données d'entraînement donne le point de variance minimale celui-ci pourrait ne pas correspondre à un cluster de données d'apprentissage. Ce qui n'est pas forcément très satisfaisant.

8.2 Regression Linéaire Bayésienne

On note la distribution postérieur pour les paramètres w : $p(w|X, Y) \propto p(Y|X, w)p(w)$. La distribution de la prédiction est donnée par $p(y^*|x^*, \mathcal{D}) = \int p(y^*|x^*, w)p(w|\mathcal{D})dw$. En pratique, on n'a quasiment jamais de forme analytique ni pour le postérior ni pour la distribution de la prédiction. On va donc chercher à approximer ces distributions. Il y a plusieurs approches explorées

- Approche gaussienne pour $p(w|X, Y)$: voir MacKay (1992) [16].
- Méthode de Monte Carlo : échantillonner directement pour évaluer l'intégral $p(y^*|x^*, \mathcal{D})$, voir Neal (1996) [17], Hernandez-Lobato et Adams (2015) [11], Jylänki et al. (2014) [13].
- Inférence variationnelle : on cherche à minimiser la distance d'un modèle paramétrique à $p(w|X, Y)$ (au sens de la divergence KL). Cela revient à transformer ce problème en un problème d'optimisation. voir Hinton and van Camp (1993) [12], Graves (2011) [9], Blundell et al. (2015) [4].

Dans le cas de la régression logistique bayésienne, on considère le modèle suivant $s_i = Wx_i$ pour une entrée x_i. Dans le cas multi-classe, on applique un soft-max pour obtenir la prédiction $\hat{y}_{i,k} = p(y_i = k|x - i, w)$ comme suit

$$\hat{y}_{i,k} = \frac{e^{s_k}}{\sum e^{s_j}}$$

Dans le cas binaire on considère

$$\hat{y}_{i,1} = \sigma(s_1) = \frac{1}{1 + e^{-s_1}}$$

On obtient alors

$$p(y|X,w) = \prod_{i=1}^{N} p(y_i = 1|x_i, w)$$

qui n'est plus gaussien. Or

$$p(w|X,Y) \propto p(Y|X,w)p(w)$$

n'admet donc pas de forme analytique.

8.2.1 Approximation de Laplace

On va présenter l'approche gaussienne par le biais de l'approximation de Laplace pour $p(w|X,Y)$. On approxime la distribution $p(w|X,Y)$ par une distribution $q(w) = \mathcal{N}(w;\mu,\Sigma)$. Pour se faire on commence par adapter μ au mode de la distribution définit par $\nabla_w p(w) = 0$ (en pratique on va procéder par méthode de gradient pour maximiser le log postérior $\mu = w_{\text{MAP}}$).
On fixe ensuite l'inverse de la variance à la Hessienne

$$\Sigma^{-1} = \nabla\nabla_w p(w|X,Y)\Big|_{w=\mu}$$

Cette méthode a des limites assez clair, typiquement elle ignore les propriétés globales. On se retrouve tout de même avec l'approximation suivante

$$p(y^*|x^*,\mathcal{D}) \simeq \int p(y^*|x^*,w)q(w)dw$$

Cette distribution postérior est incalculable en pratique. Deux options s'offrent à nous

— Méthode de Monte Carlo : on pose $p(y^* = 1|x^*,w) = \sigma(w^T x^*)$

$$p(y^* = 1|x^*,\mathcal{D}) \simeq \sum_{s=1}^{S} \sigma\left((w^s)^T x^*\right), \quad w^s \sim q(w)$$

— Convolution de la sigmoïde avec une gaussienne : on utilise un modèle probit, $\sigma(a) \simeq \Phi(\lambda a)$ avec $\lambda^2 = \pi/8$

$$p(y^*|x^*,\mathcal{D}) \simeq \Phi\left(\frac{\mu^T x^*}{\sqrt{\lambda^{-2} + x^{*T}\Sigma x^*}}\right)$$

8.3 Réseaux Bayésiens

Contrairement à un réseau standard, un réseau bayésien prédit une distribution $p(y_i|x_i,\mathcal{D})$. On définit

— le prior : sur les paramètres $p(w) = \mathcal{N}(w;0,\alpha^{-1}\text{Id})$

— la vraisemblance : $p(y_i|x_i,w) = \mathcal{N}(y_i; f^w(x_i),\beta^{-1})$ (pour une régression)

On cherche à calculer le postérior

$$p(w|X,Y) = \prod_{i=1}^{N} p(w|x_i, y_i, \beta) \propto p(w) \prod_{i=1}^{N} p(y_i|x_i, w)$$

, ce n'est pas une distribution gaussienne. Comme précédemment

$$p(y^*|x^*, \mathcal{D}) = \int p(y^*|x^*, w) p(w|\mathcal{D}) dw$$

ne peut pas être évaluée analytiquement. Par MC (Monte Carlo) on fait l'approximation

$$p(y^*|x^*, \mathcal{D}) = \frac{1}{S} \sum_{s=1}^{S} p(y^*|x^*, w^s), \quad w^s \sim p(w|\mathcal{D})$$

Pour sampler $p(w|\mathcal{D})$ par MCMC (Markov Chain Monte Carlo). Cependant cette méthode requiert tout le dataset et donc ne marche pas bien sur les gros datasets.
Avec la méthode d'inférence variationnelle, on cherche à approximer le psotérior $p(w|X,Y)$ en minimsant la divergence KL avec $q_\theta(w)$

$$KL(q_\theta(w) \| p(w|X,Y)) = \int q_\theta(w) \log \left(\frac{q_\theta(w)}{p(w|X,Y)} \right) dw$$

On aura alors

$$p(y^*|x^*, X, Y) = \int p(y^*|x^*, w) q_{\theta^*}(w) dw$$

Chaque poids du réseau admet sa propre moyenne μ_j et variance σ_j. On réécrit la divergence KL

$$\begin{aligned} KL(q_\theta(w) \| p(w|X,Y)) &= -\int q_\theta(w) \log \left(\frac{p(w|X,Y)}{q_\theta(w)} \right) dw \\ &= -\int q_\theta(w) \log \left(\frac{p(Y|X,w)p(w)}{q_\theta(w)p(Y|X)} \right) dw \\ = -\int q_\theta(w) \log \left(p(Y|X,w) \right) dw &+ \int q_\theta(w) \log \left(\frac{q_\theta(w)}{p(w)} \right) dw + \log(p(Y|X)) \\ = -\int q_\theta(w) \log \left(p(Y|X,w) \right) dw &+ KL(q_\theta(w) \| p(w)) + \log(p(Y|X)) \end{aligned}$$

on retrouve la Evidence Lower Bound (ELBO) : $\mathcal{L}_{\text{VI}}$

$$KL(q_\theta(w) \| p(w|X,Y)) = -\mathcal{L}_{\text{VI}}(X, Y, \theta) + \log(p(Y|X))$$

Et donc minimiser $KL(q_\theta(w) \| p(w|X,Y))$ revient à maximiser $\mathcal{L}_{\text{VI}}(X, Y, \theta)$ dont le premier terme encourage q_θ à expliquer correctement les données et le second terme encourage q_θ à rester relativement proche du prior.

8.3.1 Entraîner un BNN

Pour entraîner un BNN on doit calculer les dérivées de la ELBO par rapport au paramètres θ

$$\mathcal{L}_{\text{VI}}(X, Y, \theta) = \sum_{i=1}^{N} \int q_\theta(w) \log\left(p(y_i|f^w(x_i))\right) dw - KL(q_\theta(w)\|p(w))$$

$KL(q_\theta(w)\|p(w))$ peut souvent être traité de façon analytique. En revanche, $\mathbb{E}_{q_\theta(w)}\left[\log(p(Y|X,w))\right]$ doit être approximé sur l'ensemble du dataset. La méthode est décrite dans la figure 8.

Algorithm 1 Minimise divergence between $q_\theta(\boldsymbol{\omega})$ and $p(\boldsymbol{\omega}|X,Y)$

1: Given dataset $\mathbf{X}, \mathbf{Y}$,
2: Define learning rate schedule η,
3: Initialise parameters θ randomly.
4: **repeat**
5: Sample M random variables $\widehat{\boldsymbol{\epsilon}}_i \sim p(\boldsymbol{\epsilon})$, S a random subset of $\{1, .., N\}$ of size M.
6: Calculate stochastic derivative estimator w.r.t. θ:

$$\widehat{\Delta\theta} \leftarrow -\frac{N}{M}\sum_{i\in S}\frac{\partial}{\partial\theta}\log p(\mathbf{y}_i|\mathbf{f}^{g(\theta,\widehat{\boldsymbol{\epsilon}}_i)}(\mathbf{x}_i)) + \frac{\partial}{\partial\theta}\text{KL}(q_\theta(\boldsymbol{\omega})||p(\boldsymbol{\omega})).$$

7: Update θ:
$\theta \leftarrow \theta + \eta\widehat{\Delta\theta}$.
8: **until** θ has converged.

FIGURE 8 – algorithme de minimisation de $\mathcal{L}_{\text{VI}}(X, Y, \theta)$

8.4 Monte Carlo Dropout

Le dropout est une technique de régularisation introduite par Hinton en 2012 qui consiste à aléatoirement (avec probabilité p souvent choisie égale à 0.5) omettre certains neurones. Cette méthode a plusieurs avantages

— réduit les risque d'over-fitting (meilleure généralisation).

— empêche la co-adaptation (une des principales causes de l'over-fitting, cela correspond au fait que les neurones sont très inter-dépendants).

— correspond à moyenner plusieurs réseaux

Le dropout a tendance à ralentir la convergence. Dans le cas de l'inférence variationnelle le dropout correspond à mettre à 0 certaines lignes des matrices $W_1, ..., W_H$ aléatoirement. On note $\hat{W}$ la matrice W après dropout. On note ainsi $\hat{W} = \text{diag}(\hat{e})W$ et $W = \text{diag}(\hat{e})M$

avec $\hat{e}$ le vecteur de Bernoulli qui correspond au dropout.

$$\frac{1}{S}\sum_{s=1}^{S} p(y_i|f^{\hat{W}}(x_i)) \simeq \int p(y^*|f^W(x^*))q(W)dw$$

Dans le cas de l'entraînement pour une régression, on note $\hat{\mathcal{L}}_{\text{dropout}}$ la fonction objective définie par

$$\hat{\mathcal{L}}_{\text{dropout}}(M_1, M_2) = \frac{1}{M}\sum_{i=1}^{S} \|f^{\hat{W}}(x_i) - y_i\|^2 + \lambda_1\|M_1\|^2 + \lambda_2\|M_2\|^2$$

Dans le cas d'une vraisemblance gaussienne de variance τ^{-1}, on a

$$\|f^{\hat{W}}(x_i) - y_i\|^2 = -\frac{1}{\tau}\log\left(p(y_i|f^W(x_i))\right)$$

et donc

$$\hat{\mathcal{L}}_{\text{dropout}}(M_1, M_2) = \frac{1}{M\tau}\sum_{i=1}^{S} \log\left(p(y_i|f^W(x_i))\right) + \lambda_1\|M_1\|^2 + \lambda_2\|M_2\|^2$$

Ce problème est similaire a celui résolu par l'algorithme présenté dans la figure 8. Ainsi un réseau entraîné avec dropout peut être vu comme une approximation variationnelle bayésienne. On remplace dans l'algorithme

$$\frac{\partial}{\partial M}KL(q(W)\|p(W)) = \frac{\partial}{\partial M}N\tau(\lambda_1\|M_1\|^2 + \lambda_2\|M_2\|^2)$$

On a alors deux résultats sur la stabilité des résultats

Proposition 45. *Étant donné la distribution de prédiction* $p(y^*|f^W(x^*)) = \mathcal{N}(y^*; f^W(x^*), \tau^{-1}Id)$ *pour* $\tau > 0$. *Avec* $\hat{W}_t \simeq q(W)$ *on peut estimer en utilisant l'estimateur sans biais :*

$$\tilde{\mathbb{E}}\left[y^*\right] = \frac{1}{T}\sum_{t=1}^{T} f^{\hat{W}_t}(x^*) \underset{T\to\infty}{\longrightarrow} \mathbb{E}_{q_w^*(y^*|x^*)}\left[y^*\right]$$

C'est équivalent à effectuer T passage aléatoire par le réseau et de moyenner les résultats.

Proposition 46. *Étant donné la distribution de prédiction* $p(y^*|f^W(x^*)) = \mathcal{N}(y^*; f^W(x^*), \tau^{-1}Id)$ *pour* $\tau > 0$. *Avec* $\hat{W}_t \simeq q(W)$ *on peut estimer en utilisant l'estimateur sans biais :*

$$\tilde{\mathbb{E}}\left[(y^*)^T(y^*)\right] = \tau - Id + \frac{1}{T}\sum_{t=1}^{T} f^{\hat{W}_t}(x^*)^T f^{\hat{W}_t}(x^*) \underset{T\to\infty}{\longrightarrow} \mathbb{E}_{q_w^*(y^*|x^*)}\left[(y^*)^T(y^*)\right]$$

8.5 Incertitude en Classification

On considère trois approches pour estimer l'incertitude

— ratio de variation : En T passage, on retient la fréquence $f_x^{c^*}$ du label le plus fréquent c^*, on a alors

$$\text{var-ratio}[x] = 1 - f_x^{c^*}/T$$

— entropie prédictive : mesure la quantité moyenne d'information contenu dans la distribution prédictive

$$\hat{\mathcal{H}}\left[y|x, \mathcal{D}_{\text{train}}\right] = -\sum_c \left(\frac{1}{T}\sum_t p(y=c|x,\hat{w}_t)\right) \log\left(\frac{1}{T}\sum_t p(y=c|x,\hat{w}_t)\right)$$

— information mutuelle : maximise les informations mutuelles sur les points où le modèle est en moyenne incertain

$$\hat{\mathcal{I}}\left[y, w|x, \mathcal{D}_{\text{train}}\right] = \hat{\mathcal{H}}\left[y|x, \mathcal{D}_{\text{train}}\right] + \frac{1}{T}\sum_{c,t} p(y=c|x,\hat{w}_t)\log(p(y=c|x,\hat{w}_t))$$

On donne plusieurs exemples concrets

— toutes les prédictions sont identiques et valent 1 : $(1,0), ..., (1,0)$

$$\text{var-ratio}[x] = 0 \qquad \hat{\mathcal{H}}\left[y|x, \mathcal{D}_{\text{train}}\right] = 0 \qquad \hat{\mathcal{I}}\left[y, w|x, \mathcal{D}_{\text{train}}\right] = 0$$

— toutes les prédictions sont identiques et valent 0.5 : $(0.5, 0.5), ..., (0.5, 0.5)$

$$\text{var-ratio}[x] = 0.5 \qquad \hat{\mathcal{H}}\left[y|x, \mathcal{D}_{\text{train}}\right] = 0.5 \qquad \hat{\mathcal{I}}\left[y, w|x, \mathcal{D}_{\text{train}}\right] = 0$$

— les prédictions sont différentes et valent 1 : $(1,0), ..., (0,1)$

$$\text{var-ratio}[x] = 0.5 \qquad \hat{\mathcal{H}}\left[y|x, \mathcal{D}_{\text{train}}\right] = 0.5 \qquad \hat{\mathcal{I}}\left[y, w|x, \mathcal{D}_{\text{train}}\right] = 0.5$$

8.5.1 Échec de Prédiction

On utilise un score de confiance pour accepter ou rejeter une prédiction. Si on note la prédiction

$$\hat{y} = \arg\max_{k \in \mathcal{Y}} p(Y = k|w, x)$$

On a alors plusieurs approches pour estimer le score de confiance. En première approche on peut simplement prendre $MCP(x) = \max_{k\in\mathcal{Y}} p(Y = k|w, x)$. Le dropout de Monte Carlo est une autre approche plus avancée.
Pour construire un score de confiance $\hat{C}$, l'idéal serait d'avoir

$$p(\hat{Y} = Y|\hat{C} = p) = p$$

Cela peut être évaluer avec une diagramme de fiabilité calculé comme suit

$$\forall B_m, \quad \text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}_{\hat{y}_i = y_i} \text{ et } \text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i$$

En cas de calibration parfaite ou aura $\text{acc}(B_m) = \text{conf}(B_m)$. Certains résultats récents montrent que les réseaux récents ne sont pas bien calibrés, voir [10]. Et dans le cas de prédiction en excès de confiance une solution est de prendre le scaling suivant

$$P(\hat{y}_k) = \frac{e^{\frac{s_k}{\tau}}}{\sum_{k'=1}^{K} e^{\frac{s'_k}{\tau}}}$$

τ est optimisé sur l'ensemble de validation pour se rapprocher de $\text{acc}(B_m) = \text{conf}(B_m)$. Une autre approche consiste à apprendre cette confiance par le biais d'un réseaux de neurones. On peut alors utiliser la fonction de coût suivante

$$\mathcal{L}_{\text{conf}}(\theta, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{c}(x_i, \theta) - c^*(x_i, y_i^*)\right)^2$$

Ces méthodes de calcul de score de confiance a plusieurs applications dont par exemple

- Active Learning : cela consiste à apprendre à partir de peu de données annotées et plusieurs données sans annotation. Les scores de confiances permettent de meilleurs performances, voir [7].
- Reinforcement Learning : on peut utiliser échantillonnage de Thomspon pour aider un agent à choisir s'il doit exploiter les récompenses ou explorer son environnement, voir [6].

8.6 Autres Problèmes de Robustesse

On a vu que les approches théoriques classiques de learning ne sont pas suffisantes pour justifier de la capacité de généralisation des réseaux. La complexité de Rademacher est une première approche

$$\mathcal{R}_n(\mathcal{H}) = E_\sigma \left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i) \right]$$

détaillée dans [14], pour une visualisation voir figure 9. Dans le cas $\mathcal{R}_n(\mathcal{H}) \simeq 1$ on n'a pas de garantie sur la capacité de généralisation.

8.6.1 Stabilité des Prédictions

On pourrait souhaiter que les prédictions d'un réseaux soient invariantes par très faibles modifications. Cet aspect a été mis en avant avec l'arrivée des attaques adversaire qui consiste à entraîner un réseau à créer le plus petit masque sur les données induisant une erreur de prédiction. Ces méthodes peuvent également servir à augmenter la stabilité si utilisées pendant l'entraînement. Outre la génération de masque, on peut vouloir ajouter des occlusions partielles ou des artefacts aux quels le réseau devrait être invariant. Pour plus de détails voir [3, 19, 15, 18].

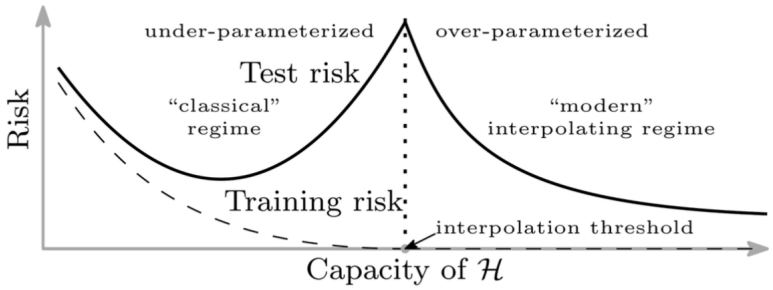

FIGURE 9 – phénomène de la courbe en double u

Références

[1] Martin Anthony and Peter L Bartlett. *Neural network learning : Theoretical foundations*. cambridge university press, 2009.

[2] Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63) :1–17, 2019.

[3] Alberto Bietti and Julien Mairal. Invariance and stability of deep convolutional representations. In *Advances in neural information processing systems*, pages 6210–6220, 2017.

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv :1505.05424*, 2015.

[5] Michel Coste. *An introduction to semialgebraic geometry*. Citeseer, 2000.

[6] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation : Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[7] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1183–1192. JMLR. org, 2017.

[8] Christophe Giraud. Fondements mathématiques de l'apprentissage statistique.

[9] Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

[11] José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015.

[12] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993.

[13] Pasi Jylänki, Aapo Nummenmaa, and Aki Vehtari. Expectation propagation for neural networks with sparsity-promoting priors. *The Journal of Machine Learning Research*, 15(1) :1849–1901, 2014.

[14] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org, 2017.

[15] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv :1610.02242*, 2016.

[16] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3) :448–472, 1992.

[17] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[18] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in neural information processing systems*, pages 1163–1171, 2016.

[19] Antti Tarvainen and Harri Valpola. Mean teachers are better role models : Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[20] Dmitry Yarotsky. Optimal approximation of continuous functions by very deep relu networks. *arXiv preprint arXiv :1802.03620*, 2018.